

## GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES

### Cross-Reference to Related Applications

10 This application is related to U.S. Patent Application Serial No. 10/213,555, filed on August 7, 2002, entitled IMAGE DISPLAY SYSTEM AND METHOD; U.S. Patent Application Serial No. 10/242,195, filed on September 11, 2002, entitled IMAGE DISPLAY SYSTEM AND METHOD; U.S. Patent Application Serial No. 10/242,545, filed on September 11, 2002, entitled IMAGE DISPLAY SYSTEM AND METHOD; U.S. Patent Application Serial No. 10/631,681, filed July 31, 2003, entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES; U.S. Patent Application Serial No. 10/632,042, filed July 31, 2003, entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES; U.S. Patent Application Serial No. 10/672,845, filed September 26, 2003, entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES; U.S. Patent Application Serial No. 10/672,544, filed September 26, 2003, entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES; U.S. Patent Application Serial No. 10/697,605, filed October 30, 2003, and entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES ON A DIAMOND GRID; U.S. Patent Application Serial No. 10/696,888, filed October 30, 2003, and entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES ON DIFFERENT TYPES OF GRIDS; U.S. Patent Application Serial No. 10/697,830, filed October 30, 2003, and entitled IMAGE DISPLAY SYSTEM AND METHOD; U.S. Patent Application Serial No. 10/750,591, filed December 31, 2003, entitled DISPLAYING SPATIALLY OFFSET SUB-FRAMES WITH A

DISPLAY DEVICE HAVING A SET OF DEFECTIVE DISPLAY PIXELS; U.S. Patent Application Serial No. 10/768,621, filed January 30, 2004, entitled GENERATING AND DISPLAYING SPATIALLY OFFSET SUB-FRAMES; and U.S. Patent Application Serial No. 10/768,215, filed January 30, 2004, entitled

5 DISPLAYING SUB-FRAMES AT SPATIALLY OFFSET POSITIONS ON A CIRCLE. Each of the above U.S. Patent Applications is assigned to the assignee of the present invention, and is hereby incorporated by reference herein.

10 **Background**

A conventional system or device for displaying an image, such as a display, projector, or other imaging system, produces a displayed image by addressing an array of individual picture elements or pixels arranged in 15 horizontal rows and vertical columns. A resolution of the displayed image is defined as the number of horizontal rows and vertical columns of individual pixels forming the displayed image. The resolution of the displayed image is affected by a resolution of the display device itself as well as a resolution of the image data processed by the display device and used to produce the displayed 20 image.

Typically, to increase a resolution of the displayed image, the resolution of the display device as well as the resolution of the image data used to produce the displayed image must be increased. Increasing a resolution of the display device, however, increases a cost and complexity of the display device. In 25 addition, higher resolution image data may not be available and/or may be difficult to generate.

It would be desirable to be able to enhance the display of various types of graphical images including natural images and high contrast images such as business graphics.

## Summary

One form of the present invention provides a method of displaying an image with a display device comprises receiving image data for the image, 5 generating first and second sub-frames where the first and the second sub-frames comprise a plurality of sub-frame pixel values and a plurality of error values and where at least a first one of the plurality of sub-frame pixel values is calculated using the image data, at least a second one of the plurality of sub-frame pixel values, and at least one of the plurality of error values, and 10 alternating between displaying the first sub-frame in a first position and displaying the second sub-frame in a second position spatially offset from the first position.

## Brief Description of the Drawings

15       Figure 1 is a block diagram illustrating an image display system according to one embodiment of the present invention.

      Figures 2A-2C are schematic diagrams illustrating the display of two sub-frames according to one embodiment of the present invention.

20       Figures 3A-3E are schematic diagrams illustrating the display of four sub-frames according to one embodiment of the present invention.

      Figures 4A-4E are schematic diagrams illustrating the display of a pixel with an image display system according to one embodiment of the present invention.

25       Figure 5 is a diagram illustrating the generation of low resolution sub-frames from an original high resolution image using a nearest neighbor algorithm according to one embodiment of the present invention.

      Figure 6 is a diagram illustrating the generation of low resolution sub-frames from an original high resolution image using a bilinear algorithm

30       according to one embodiment of the present invention.

      Figure 7 is a block diagram illustrating a system for generating a simulated high resolution image according to one embodiment of the present invention.

Figure 8 is a block diagram illustrating a system for generating a simulated high resolution image for two-position processing based on separable upsampling according to one embodiment of the present invention.

5 Figure 9 is a block diagram illustrating a system for generating a simulated high resolution image for two-position processing based on non-separable upsampling according to one embodiment of the present invention.

Figure 10 is a block diagram illustrating a system for generating a simulated high resolution image for four-position processing according to one embodiment of the present invention.

10 Figure 11 is a block diagram illustrating the comparison of a simulated high resolution image and a desired high resolution image according to one embodiment of the present invention.

Figure 12 is a diagram illustrating the effect in the frequency domain of the upsampling of a sub-frame according to one embodiment of the present invention.

15 Figure 13 is a diagram illustrating the effect in the frequency domain of the shifting of an upsampled sub-frame according to one embodiment of the present invention.

Figure 14 is a diagram illustrating regions of influence for pixels in an upsampled image according to one embodiment of the present invention.

20 Figure 15 is a diagram illustrating the generation of an initial simulated high resolution image based on an adaptive multi-pass algorithm according to one embodiment of the present invention.

Figure 16 is a diagram illustrating the generation of correction data based on an adaptive multi-pass algorithm according to one embodiment of the present invention.

25 Figure 17 is a diagram illustrating the generation of updated sub-frames based on an adaptive multi-pass algorithm according to one embodiment of the present invention.

30 Figure 18 is a diagram illustrating the generation of correction data based on an adaptive multi-pass algorithm according to another embodiment of the present invention.

Figures 19A-19E are schematic diagrams illustrating the display of four sub-frames with respect to an original high resolution image according to one embodiment of the present invention.

Figure 20 is a block diagram illustrating a system for generating a simulated high resolution image for four-position processing using a center adaptive multi-pass algorithm according to one embodiment of the present invention.

Figure 21 is a block diagram illustrating the generation of correction data using a center adaptive multi-pass algorithm according to one embodiment of the present invention.

Figure 22 is a block diagram illustrating a system for generating a simulated high resolution image for four-position processing using a simplified center adaptive multi-pass algorithm according to one embodiment of the present invention.

Figure 23 is a block diagram illustrating the generation of correction data using a simplified center adaptive multi-pass algorithm according to one embodiment of the present invention.

Figures 24A-24C are block diagrams illustrating regions of influence for a pixel for different numbers of iterations of the adaptive multi-pass algorithm according to one embodiment of the present invention.

Figure 25 is a block diagram illustrating a region of influence of a pixel with respect to an image according to one embodiment of the present invention.

Figure 26 is a block diagram illustrating calculated history values in a region of influence of a pixel according to one embodiment of the present invention.

Figure 27 is a block diagram illustrating calculated history values in a simplified region of influence of a pixel according to one embodiment of the present invention.

Figure 28 is a block diagram illustrating a simplified region of influence of a pixel with respect to an image according to one embodiment of the present invention.

Figure 29 is a block diagram illustrating portions of a sub-frame generation unit according to one embodiment of the present invention.

Figure 30 is a block diagram illustrating intertwined sub-frames for two position processing.

5       Figure 31 is a block diagram illustrating calculated history and error values in a simplified region of influence of a pixel according to one embodiment of the present invention.

Figure 32 is a block diagram illustrating a simplified region of influence of a pixel with respect to an image according to one embodiment of the present  
10 invention.

### Detailed Description

15       In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing  
20 from the scope of the present invention. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims.

#### I. Spatial and Temporal Shifting of Sub-frames

25       Some display systems, such as some digital light projectors, may not have sufficient resolution to display some high resolution images. Such systems can be configured to give the appearance to the human eye of higher resolution images by displaying spatially and temporally shifted lower resolution images. The lower resolution images are referred to as sub-frames. A problem  
30 of sub-frame generation, which is addressed by embodiments of the present invention, is to determine appropriate values for the sub-frames so that the

displayed sub-frames are close in appearance to how the high-resolution image from which the sub-frames were derived would appear if directly displayed.

One embodiment of a display system that provides the appearance of enhanced resolution through temporal and spatial shifting of sub-frames is

5 described in the above-cited U.S. patent applications, and is summarized below with reference to Figures 1-4E.

Figure 1 is a block diagram illustrating an image display system 10 according to one embodiment of the present invention. Image display system 10 facilitates processing of an image 12 to create a displayed image 14. Image 10 12 is defined to include any pictorial, graphical, and/or textural characters, symbols, illustrations, and/or other representation of information. Image 12 is represented, for example, by image data 16. Image data 16 includes individual picture elements or pixels of image 12. While one image is illustrated and described as being processed by image display system 10, it is understood that 15 a plurality or series of images may be processed and displayed by image display system 10.

In one embodiment, image display system 10 includes a frame rate conversion unit 20 and an image frame buffer 22, an image processing unit 24, and a display device 26. As described below, frame rate conversion unit 20 and 20 image frame buffer 22 receive and buffer image data 16 for image 12 to create an image frame 28 for image 12. Image processing unit 24 processes image frame 28 to define one or more image sub-frames 30 for image frame 28, and display device 26 temporally and spatially displays image sub-frames 30 to produce displayed image 14.

25 Image display system 10, including frame rate conversion unit 20 and/or image processing unit 24, includes hardware, software, firmware, or a combination of these. In one embodiment, one or more components of image display system 10, including frame rate conversion unit 20 and/or image processing unit 24, are included in a computer, computer server, or other

30 microprocessor-based system capable of performing a sequence of logic operations. In addition, processing can be distributed throughout the system with individual portions being implemented in separate system components.

Image data 16 may include digital image data 161 or analog image data 162. To process analog image data 162, image display system 10 includes an analog-to-digital (A/D) converter 32. As such, A/D converter 32 converts analog image data 162 to digital form for subsequent processing. Thus, image display system 10 may receive and process digital image data 161 and/or analog image data 162 for image 12.

Frame rate conversion unit 20 receives image data 16 for image 12 and buffers or stores image data 16 in image frame buffer 22. More specifically, frame rate conversion unit 20 receives image data 16 representing individual lines or fields of image 12 and buffers image data 16 in image frame buffer 22 to create image frame 28 for image 12. Image frame buffer 22 buffers image data 16 by receiving and storing all of the image data for image frame 28, and frame rate conversion unit 20 creates image frame 28 by subsequently retrieving or extracting all of the image data for image frame 28 from image frame buffer 22. As such, image frame 28 is defined to include a plurality of individual lines or fields of image data 16 representing an entirety of image 12. Thus, image frame 28 includes a plurality of columns and a plurality of rows of individual pixels representing image 12.

Frame rate conversion unit 20 and image frame buffer 22 can receive and process image data 16 as progressive image data and/or interlaced image data. With progressive image data, frame rate conversion unit 20 and image frame buffer 22 receive and store sequential fields of image data 16 for image 12. Thus, frame rate conversion unit 20 creates image frame 28 by retrieving the sequential fields of image data 16 for image 12. With interlaced image data, frame rate conversion unit 20 and image frame buffer 22 receive and store odd fields and even fields of image data 16 for image 12. For example, all of the odd fields of image data 16 are received and stored and all of the even fields of image data 16 are received and stored. As such, frame rate conversion unit 20 de-interlaces image data 16 and creates image frame 28 by retrieving the odd and even fields of image data 16 for image 12.

Image frame buffer 22 includes memory for storing image data 16 for one or more image frames 28 of respective images 12. Thus, image frame buffer 22

constitutes a database of one or more image frames 28. Examples of image frame buffer 22 include non-volatile memory (e.g., a hard disk drive or other persistent storage device) and may include volatile memory (e.g., random access memory (RAM)).

5 By receiving image data 16 at frame rate conversion unit 20 and buffering image data 16 with image frame buffer 22, input timing of image data 16 can be decoupled from a timing requirement of display device 26. More specifically, since image data 16 for image frame 28 is received and stored by image frame buffer 22, image data 16 can be received as input at any rate. As such, the  
10 frame rate of image frame 28 can be converted to the timing requirement of display device 26. Thus, image data 16 for image frame 28 can be extracted from image frame buffer 22 at a frame rate of display device 26.

In one embodiment, image processing unit 24 includes a resolution adjustment unit 34 and a sub-frame generation unit 36. As described below,  
15 resolution adjustment unit 34 receives image data 16 for image frame 28 and adjusts a resolution of image data 16 for display on display device 26, and sub-frame generation unit 36 generates a plurality of image sub-frames 30 for image frame 28. More specifically, image processing unit 24 receives image data 16 for image frame 28 at an original resolution and processes image data 16 to  
20 increase, decrease, and/or leave unaltered the resolution of image data 16. Accordingly, with image processing unit 24, image display system 10 can receive and display image data 16 of varying resolutions.

Sub-frame generation unit 36 receives and processes image data 16 for image frame 28 to define a plurality of image sub-frames 30 for image frame 28.  
25 If resolution adjustment unit 34 has adjusted the resolution of image data 16, sub-frame generation unit 36 receives image data 16 at the adjusted resolution. The adjusted resolution of image data 16 may be increased, decreased, or the same as the original resolution of image data 16 for image frame 28. Sub-frame generation unit 36 generates image sub-frames 30 with a resolution which  
30 matches the resolution of display device 26. Image sub-frames 30 are each of an area equal to image frame 28. Sub-frames 30 each include a plurality of columns and a plurality of rows of individual pixels representing a subset of

image data 16 of image 12, and have a resolution that matches the resolution of display device 26.

Each image sub-frame 30 includes a matrix or array of pixels for image frame 28. Image sub-frames 30 are spatially offset from each other such that 5 each image sub-frame 30 includes different pixels and/or portions of pixels. As such, image sub-frames 30 are offset from each other by a vertical distance and/or a horizontal distance, as described below.

Display device 26 receives image sub-frames 30 from image processing unit 24 and sequentially displays image sub-frames 30 to create displayed 10 image 14. More specifically, as image sub-frames 30 are spatially offset from each other, display device 26 displays image sub-frames 30 in different positions according to the spatial offset of image sub-frames 30, as described below. As such, display device 26 alternates between displaying image sub-frames 30 for image frame 28 to create displayed image 14. Accordingly, 15 display device 26 displays an entire sub-frame 30 for image frame 28 at one time.

In one embodiment, display device 26 performs one cycle of displaying image sub-frames 30 for each image frame 28. Display device 26 displays image sub-frames 30 so as to be spatially and temporally offset from each 20 other. In one embodiment, display device 26 optically steers image sub-frames 30 to create displayed image 14. As such, individual pixels of display device 26 are addressed to multiple locations.

In one embodiment, display device 26 includes an image shifter 38. Image shifter 38 spatially alters or offsets the position of image sub-frames 30 25 as displayed by display device 26. More specifically, image shifter 38 varies the position of display of image sub-frames 30, as described below, to produce displayed image 14.

In one embodiment, display device 26 includes a light modulator for modulation of incident light. The light modulator includes, for example, a 30 plurality of micro-mirror devices arranged to form an array of micro-mirror devices. As such, each micro-mirror device constitutes one cell or pixel of

display device 26. Display device 26 may form part of a display, projector, or other imaging system.

In one embodiment, image display system 10 includes a timing generator 40. Timing generator 40 communicates, for example, with frame rate conversion unit 20, image processing unit 24, including resolution adjustment unit 34 and sub-frame generation unit 36, and display device 26, including image shifter 38. As such, timing generator 40 synchronizes buffering and conversion of image data 16 to create image frame 28, processing of image frame 28 to adjust the resolution of image data 16 and generate image sub-frames 30, and positioning and displaying of image sub-frames 30 to produce displayed image 14. Accordingly, timing generator 40 controls timing of image display system 10 such that entire sub-frames of image 12 are temporally and spatially displayed by display device 26 as displayed image 14.

In one embodiment, as illustrated in Figures 2A and 2B, image processing unit 24 defines two image sub-frames 30 for image frame 28. More specifically, image processing unit 24 defines a first sub-frame 301 and a second sub-frame 302 for image frame 28. As such, first sub-frame 301 and second sub-frame 302 each include a plurality of columns and a plurality of rows of individual pixels 18 of image data 16. Thus, first sub-frame 301 and second sub-frame 302 each constitute an image data array or pixel matrix of a subset of image data 16.

In one embodiment, as illustrated in Figure 2B, second sub-frame 302 is offset from first sub-frame 301 by a vertical distance 50 and a horizontal distance 52. As such, second sub-frame 302 is spatially offset from first sub-frame 301 by a predetermined distance. In one illustrative embodiment, vertical distance 50 and horizontal distance 52 are each approximately one-half of one pixel.

As illustrated in Figure 2C, display device 26 alternates between displaying first sub-frame 301 in a first position and displaying second sub-frame 302 in a second position spatially offset from the first position. More specifically, display device 26 shifts display of second sub-frame 302 relative to display of first sub-frame 301 by vertical distance 50 and horizontal distance 52. As such,

pixels of first sub-frame 301 overlap pixels of second sub-frame 302. In one embodiment, display device 26 performs one cycle of displaying first sub-frame 301 in the first position and displaying second sub-frame 302 in the second position for image frame 28. Thus, second sub-frame 302 is spatially and

5 temporally displayed relative to first sub-frame 301. The display of two temporally and spatially shifted sub-frames in this manner is referred to herein as two-position processing.

In another embodiment, as illustrated in Figures 3A-3D, image processing unit 24 defines four image sub-frames 30 for image frame 28. More 10 specifically, image processing unit 24 defines a first sub-frame 301, a second sub-frame 302, a third sub-frame 303, and a fourth sub-frame 304 for image frame 28. As such, first sub-frame 301, second sub-frame 302, third sub-frame 303, and fourth sub-frame 304 each include a plurality of columns and a plurality of rows of individual pixels 18 of image data 16.

15 In one embodiment, as illustrated in Figures 3B-3D, second sub-frame 302 is offset from first sub-frame 301 by a vertical distance 50 and a horizontal distance 52, third sub-frame 303 is offset from first sub-frame 301 by a horizontal distance 54, and fourth sub-frame 304 is offset from first sub-frame 301 by a vertical distance 56. As such, second sub-frame 302, third sub-frame 20 303, and fourth sub-frame 304 are each spatially offset from each other and spatially offset from first sub-frame 301 by a predetermined distance. In one illustrative embodiment, vertical distance 50, horizontal distance 52, horizontal distance 54, and vertical distance 56 are each approximately one-half of one pixel.

25 As illustrated schematically in Figure 3E, display device 26 alternates between displaying first sub-frame 301 in a first position  $P_1$ , displaying second sub-frame 302 in a second position  $P_2$  spatially offset from the first position, displaying third sub-frame 303 in a third position  $P_3$  spatially offset from the first position, and displaying fourth sub-frame 304 in a fourth position  $P_4$  spatially 30 offset from the first position. More specifically, display device 26 shifts display of second sub-frame 302, third sub-frame 303, and fourth sub-frame 304 relative to first sub-frame 301 by the respective predetermined distance. As

such, pixels of first sub-frame 301, second sub-frame 302, third sub-frame 303, and fourth sub-frame 304 overlap each other.

In one embodiment, display device 26 performs one cycle of displaying first sub-frame 301 in the first position, displaying second sub-frame 302 in the 5 second position, displaying third sub-frame 303 in the third position, and displaying fourth sub-frame 304 in the fourth position for image frame 28. Thus, second sub-frame 302, third sub-frame 303, and fourth sub-frame 304 are spatially and temporally displayed relative to each other and relative to first sub-frame 301. The display of four temporally and spatially shifted sub-frames in 10 this manner is referred to herein as four-position processing.

Figures 4A-4E illustrate one embodiment of completing one cycle of displaying a pixel 181 from first sub-frame 301 in the first position, displaying a pixel 182 from second sub-frame 302 in the second position, displaying a pixel 183 from third sub-frame 303 in the third position, and displaying a pixel 184 15 from fourth sub-frame 304 in the fourth position. More specifically, Figure 4A illustrates display of pixel 181 from first sub-frame 301 in the first position, Figure 4B illustrates display of pixel 182 from second sub-frame 302 in the second position (with the first position being illustrated by dashed lines), Figure 4C illustrates display of pixel 183 from third sub-frame 303 in the third position 20 (with the first position and the second position being illustrated by dashed lines), Figure 4D illustrates display of pixel 184 from fourth sub-frame 304 in the fourth position (with the first position, the second position, and the third position being illustrated by dashed lines), and Figure 4E illustrates display of pixel 181 from first sub-frame 301 in the first position (with the second position, the third 25 position, and the fourth position being illustrated by dashed lines).

Sub-frame generation unit 36 (Figure 1) generates sub-frames 30 based on image data in image frame 28. It will be understood by a person of ordinary skill in the art that functions performed by sub-frame generation unit 36 may be implemented in hardware, software, firmware, or any combination thereof. The 30 implementation may be via a microprocessor, programmable logic device, or state machine. Components of the present invention may reside in software on one or more computer-readable mediums. The term computer-readable

medium as used herein is defined to include any kind of memory, volatile or non-volatile, such as floppy disks, hard disks, CD-ROMs, flash memory, read-only memory (ROM), and random access memory.

In one form of the invention, sub-frames 30 have a lower resolution than image frame 28. Thus, sub-frames 30 are also referred to herein as low resolution images 30, and image frame 28 is also referred to herein as a high resolution image 28. It will be understood by persons of ordinary skill in the art that the terms low resolution and high resolution are used herein in a comparative fashion, and are not limited to any particular minimum or maximum number of pixels. In one embodiment, sub-frame generation unit 36 is configured to generate sub-frames 30 based on one or more of ten algorithms. These ten algorithms are referred to herein as the following: (1) nearest neighbor; (2) bilinear; (3) spatial domain; (4) frequency domain; (5) adaptive multi-pass; (6) center adaptive multi-pass; (7) simplified center adaptive multi-pass; (8) adaptive multi-pass with history; (9) simplified center adaptive multi-pass with history; and (10) center adaptive multi-pass with history.

The nearest neighbor algorithm and the bilinear algorithm according to one form of the invention generate sub-frames 30 by combining pixels from a high resolution image 28. The spatial domain algorithm and the frequency domain algorithm according to one form of the invention generate sub-frames 30 based on the minimization of a global error metric that represents a difference between a simulated high resolution image and a desired high resolution image 28. The adaptive multi-pass algorithm, center adaptive multi-pass algorithm, simplified center adaptive multi-pass algorithm, adaptive multi-pass algorithm with history, simplified center adaptive multi-pass algorithm with history, and center adaptive multi-pass algorithm with history according to various forms of the invention generate sub-frames 30 based on the minimization of a local error metric. In one embodiment, sub-frame generation unit 36 includes memory for storing a relationship between sub-frame values and high resolution image values, wherein the relationship is based on minimization of an error metric between the high resolution image values and a simulated high resolution image that is a function of the sub-frame values.

Embodiments of each of these ten algorithms are described below with reference to Figures 5-32.

## II. Nearest Neighbor

5 Figure 5 is a diagram illustrating the generation of low resolution sub-frames 30A and 30B (collectively referred to as sub-frames 30) from an original high resolution image 28 using a nearest neighbor algorithm according to one embodiment of the present invention. In the illustrated embodiment, high 10 resolution image 28 includes four columns and four rows of pixels, for a total of sixteen pixels H1-H16. In one embodiment of the nearest neighbor algorithm, a first sub-frame 30A is generated by taking every other pixel in a first row of the high resolution image 28, skipping the second row of the high resolution image 28, taking every other pixel in the third row of the high resolution image 28, and 15 repeating this process throughout the high resolution image 28. Thus, as shown in Figure 5, the first row of sub-frame 30A includes pixels H1 and H3, and the second row of sub-frame 30A includes pixels H9 and H11. In one form of the invention, a second sub-frame 30B is generated in the same manner as the first sub-frame 30A, but the process begins at a pixel H6 that is shifted down one row and over one column from the first pixel H1. Thus, as shown in Figure 20 5, the first row of sub-frame 30B includes pixels H6 and H8, and the second row of sub-frame 30B includes pixels H14 and H16.

In one embodiment, the nearest neighbor algorithm is implemented with a 2x2 filter with three filter coefficients of "0" and a fourth filter coefficient of "1" to generate a weighted sum of the pixel values from the high resolution image.

25 Displaying sub-frames 30A and 30B using two-position processing as described above gives the appearance of a higher resolution image. The nearest neighbor algorithm is also applicable to four-position processing, and is not limited to images having the number of pixels shown in Figure 5.

## 30 III. Bilinear

Figure 6 is a diagram illustrating the generation of low resolution sub-frames 30C and 30D (collectively referred to as sub-frames 30) from an original

high resolution image 28 using a bilinear algorithm according to one embodiment of the present invention. In the illustrated embodiment, high resolution image 28 includes four columns and four rows of pixels, for a total of sixteen pixels H1-H16. Sub-frame 30C includes two columns and two rows of 5 pixels, for a total of four pixels L1-L4. And sub-frame 30D includes two columns and two rows of pixels, for a total of four pixels L5-L8.

In one embodiment, the values for pixels L1-L8 in sub-frames 30C and 30D are generated from the pixel values H1-H16 of image 28 based on the following Equations I-VIII:

10 Equation I

$$L1 = (4H1 + 2H2 + 2H5) / 8$$

Equation II

$$L2 = (4H3 + 2H4 + 2H7) / 8$$

Equation III

15  $L3 = (4H9 + 2H10 + 2H13) / 8$

Equation IV

$$L4 = (4H11 + 2H12 + 2H15) / 8$$

Equation V

$$L5 = (4H6 + 2H2 + 2H5) / 8$$

20 Equation VI

$$L6 = (4H8 + 2H4 + 2H7) / 8$$

Equation VII

$$L7 = (4H14 + 2H10 + 2H13) / 8$$

Equation VIII

25  $L8 = (4H16 + 2H12 + 2H15) / 8$

As can be seen from the above Equations I-VIII, the values of the pixels L1-L4 in sub-frame 30C are influenced the most by the values of pixels H1, H3, H9, and H11, respectively, due to the multiplication by four. But the values for the pixels L1-L4 in sub-frame 30C are also influenced by the values of diagonal 30 neighbors of pixels H1, H3, H9, and H11. Similarly, the values of the pixels L5-L8 in sub-frame 30D are influenced the most by the values of pixels H6, H8, H14, and H16, respectively, due to the multiplication by four. But the values for

the pixels L5-L8 in sub-frame 30D are also influenced by the values of diagonal neighbors of pixels H6, H8, H14, and H16.

In one embodiment, the bilinear algorithm is implemented with a 2x2 filter with one filter coefficient of "0" and three filter coefficients having a non-zero

5 value (e.g., 4, 2, and 2) to generate a weighted sum of the pixel values from the high resolution image. In another embodiment, other values are used for the filter coefficients. Displaying sub-frames 30C and 30D using two-position processing as described above gives the appearance of a higher resolution image. The bilinear algorithm is also applicable to four-position processing, and

10 is not limited to images having the number of pixels shown in Figure 6.

In one form of the nearest neighbor and bilinear algorithms, sub-frames 30 are generated based on a linear combination of pixel values from an original high resolution image as described above. In another embodiment, sub-frames 30 are generated based on a non-linear combination of pixel values from an

15 original high resolution image. For example, if the original high resolution image is gamma-corrected, appropriate non-linear combinations are used in one embodiment to undo the effect of the gamma curve.

#### IV. Systems for Generating Simulated High Resolution Images

20 Figures 7-10, 20, and 22 illustrate systems for generating simulated high resolution images. Based on these systems, spatial domain, frequency domain, adaptive multi-pass, center adaptive multi-pass, and simplified center adaptive multi-pass algorithms for generating sub-frames are developed, as described in further detail below.

25 Figure 7 is a block diagram illustrating a system 400 for generating a simulated high resolution image 412 from two 4x4 pixel low resolution sub-frames 30E according to one embodiment of the present invention. System 400 includes upsampling stage 402, shifting stage 404, convolution stage 406, and summation stage 410. Sub-frames 30E are upsampled by upsampling stage 30 402 based on a sampling matrix,  $M$ , thereby generating upsampled images. The upsampled images are shifted by shifting stage 404 based on a spatial shifting matrix,  $S$ , thereby generating shifted upsampled images. The shifted

upsampled images are convolved with an interpolating filter at convolution stage 406, thereby generating blocked images 408. In the illustrated embodiment, the interpolating filter is a 2x2 filter with filter coefficients of "1", and with the center of the convolution being the upper left position in the 2x2 matrix. The 5 interpolating filter simulates the superposition of low resolution sub-frames on a high resolution grid. The low resolution sub-frame pixel data is expanded so that the sub-frames can be represented on a high resolution grid. The interpolating filter fills in the missing pixel data produced by upsampling. The blocked images 408 are weighted and summed by summation block 410 to 10 generate the 8x8 pixel simulated high resolution image 412.

Figure 8 is a block diagram illustrating a system 500 for generating a simulated high resolution image 512 for two-position processing based on separable upsampling of two 4x4 pixel low resolution sub-frames 30F and 30G according to one embodiment of the present invention. System 500 includes 15 upsampling stages 502 and 514, shifting stage 518, convolution stages 506 and 522, summation stage 508, and multiplication stage 510. Sub-frame 30F is upscaled by a factor of two by upsampling stage 502, thereby generating an 8x8 pixel upsampled image 504. The dark pixels in upsampled image 504 represent the sixteen pixels from sub-frame 30F, and the light pixels in 20 upsampled image 504 represent zero values. Sub-frame 30G is upscaled by a factor of two by upsampling stage 514, thereby generating an 8x8 pixel upsampled image 516. The dark pixels in upsampled image 516 represent the sixteen pixels from sub-frame 30G, and the light pixels in upsampled image 516 represent zero values. In one embodiment, upsampling stages 502 and 514 25 upsample sub-frames 30F and 30G, respectively, using a diagonal sampling matrix.

The upsampled image 516 is shifted by shifting stage 518 based on a spatial shifting matrix,  $\mathbf{S}$ , thereby generating shifted upsampled image 520. In the illustrated embodiment, shifting stage 518 performs a one pixel diagonal 30 shift. Images 504 and 520 are convolved with an interpolating filter at convolution stages 506 and 522, respectively, thereby generating blocked images. In the illustrated embodiment, the interpolating filter at convolution

stages 506 and 522 is a 2x2 filter with filter coefficients of "1", and with the center of the convolution being the upper left position in the 2x2 matrix. The blocked images generated at convolution stages 506 and 522 are summed by summation block 508, and multiplied by a factor of 0.5 at multiplication stage

5 510, to generate the 8x8 pixel simulated high resolution image 512. The image data is multiplied by a factor of 0.5 at multiplication stage 510 because, in one embodiment, each of the sub-frames 30F and 30G is displayed for only half of the time slot per period allotted to a color. In another embodiment, rather than multiplying by a factor of 0.5 at multiplication stage 510, the filter coefficients of  
10 the interpolating filter at stages 506 and 522 are reduced by a factor of 0.5.

In one embodiment, as shown in Figure 8 and described above, the low resolution sub-frame data is represented by two separate sub-frames 30F and 30G, which are separately upsampled based on a diagonal sampling matrix (i.e., separable upsampling). In another embodiment, as described below with  
15 reference to Figure 9, the low resolution sub-frame data is represented by a single sub-frame, which is upsampled based on a non-diagonal sampling matrix (i.e., non-separable upsampling).

Figure 9 is a block diagram illustrating a system 600 for generating a simulated high resolution image 610 for two-position processing based on non-separable upsampling of an 8x4 pixel low resolution sub-frame 30H according to one embodiment of the present invention. System 600 includes quincunx upsampling stage 602, convolution stage 606, and multiplication stage 608. Sub-frame 30H is upsampled by quincunx upsampling stage 602 based on a quincunx sampling matrix,  $Q$ , thereby generating upsampled image 604. The  
25 dark pixels in upsampled image 604 represent the thirty-two pixels from sub-frame 30H, and the light pixels in upsampled image 604 represent zero values. Sub-frame 30H includes pixel data for two 4x4 pixel sub-frames for two-position processing. The dark pixels in the first, third, fifth, and seventh rows of  
30 upsampled image 604 represent pixels for a first 4x4 pixel sub-frame, and the dark pixels in the second, fourth, sixth, and eighth rows of upsampled image 604 represent pixels for a second 4x4 pixel sub-frame.

The upsampled image 604 is convolved with an interpolating filter at convolution stage 606, thereby generating a blocked image. In the illustrated embodiment, the interpolating filter is a 2x2 filter with filter coefficients of "1", and with the center of the convolution being the upper left position in the 2x2 matrix. The blocked image generated by convolution stage 606 is multiplied by a factor of 0.5 at multiplication stage 608, to generate the 8x8 pixel simulated high resolution image 610.

Figure 10 is a block diagram illustrating a system 700 for generating a simulated high resolution image 706 for four-position processing based on sub-frame 301 according to one embodiment of the present invention. In the embodiment illustrated in Figure 10, sub-frame 301 is an 8x8 array of pixels. Sub-frame 301 includes pixel data for four 4x4 pixel sub-frames for four-position processing. Pixels A1-A16 represent pixels for a first 4x4 pixel sub-frame, pixels B1-B16 represent pixels for a second 4x4 pixel sub-frame, pixels C1-C16 represent pixels for a third 4x4 pixel sub-frame, and pixels D1-D16 represent pixels for a fourth 4x4 pixel sub-frame.

The sub-frame 301 is convolved with an interpolating filter at convolution stage 702, thereby generating a blocked image. In the illustrated embodiment, the interpolating filter is a 2x2 filter with filter coefficients of "1", and with the center of the convolution being the upper left position in the 2x2 matrix. The blocked image generated by convolution stage 702 is multiplied by a factor of 0.25 at multiplication stage 704, to generate the 8x8 pixel simulated high resolution image 706. The image data is multiplied by a factor of 0.25 at multiplication stage 704 because, in one embodiment, each of the four sub-frames represented by sub-frame 301 is displayed for only one fourth of the time slot per period allotted to a color. In another embodiment, rather than multiplying by a factor of 0.25 at multiplication stage 704, the filter coefficients of the interpolating filter are correspondingly reduced.

30 V. Generation of Sub-frames Based on Error Minimization

As described above, systems 400, 500, 600, and 700 generate simulated high resolution images 412, 512, 610, and 706, respectively, based on low

resolution sub-frames. If the sub-frames are optimal, the simulated high resolution image will be as close as possible to the original high resolution image 28. Various error metrics may be used to determine how close a simulated high resolution image is to an original high resolution image, including

5 mean square error, weighted mean square error, as well as others.

Figure 11 is a block diagram illustrating the comparison of a simulated high resolution image 412/512/610/706 and a desired high resolution image 28 according to one embodiment of the present invention. A simulated high resolution image 412, 512, 610, or 706, is subtracted on a pixel-by-pixel basis

10 from high resolution image 28 at subtraction stage 802. In one embodiment, the resulting error image data is filtered by a human visual system (HVS) weighting filter (W) 804. In one form of the invention, HVS weighting filter 804 filters the error image data based on characteristics of the human visual system. In one embodiment, HVS weighting filter 804 reduces or eliminates high frequency  
15 errors. The mean squared error of the filtered data is then determined at stage 806 to provide a measure of how close the simulated high resolution image 412, 512, 610, or 706 is to the desired high resolution image 28.

In one embodiment, systems 400, 500, 600, and 700 are represented mathematically in an error cost equation that measures the difference between

20 a simulated high resolution image 412, 512, 610, or 706, and the original high resolution image 28. Optimal sub-frames are identified by solving the error cost equation for the sub-frame data that provides the minimum error between the simulated high resolution image and the desired high resolution image. In one embodiment, globally optimum solutions are obtained in the spatial domain and  
25 in the frequency domain, and a locally optimum solution is obtained using an adaptive multi-pass algorithm. The spatial domain, frequency domain, and adaptive multi-pass algorithms are described in further detail below with reference to Figures 12-18. The center adaptive multi-pass and simplified center adaptive multi-pass algorithms are described in further detail below with  
30 reference to Figures 19-23. The adaptive multi-pass with history, simplified center adaptive multi-pass with history, and center adaptive multi-pass with

history algorithms are described in further detail below with reference to Figures 24-32.

5 VI. Spatial Domain

A spatial domain solution for generating optimal sub-frames according to one embodiment is described in the context of the system 600 shown in Figure 9. The system 600 shown in Figure 9 can be represented mathematically in an error cost function by the following Equation IX:

10 Equation IX

$$I_Q^* = \underset{I_Q}{\operatorname{argmin}} J = \underset{I_Q}{\operatorname{argmin}} \sum_n \left( \sum_k I_Q(k) f(n - k) - h(n) \right)^2$$

where:

$I_Q^*$  = optimal low resolution data for sub-frame

30H;

$J$  = error cost function to be minimized;

15  $n$  and  $k$  = indices for identifying high resolution pixel locations for images 604 and 610;

$I_Q(k)$  = image data from upsampled image 604 at location  $k$ ;

$f(n - k)$  = filter coefficient of the interpolating filter at a position  $n - k$ ; and

20  $h(n)$  = image data for desired high resolution image 28 at location  $n$ .

The summation of " $I_Q(k) f(n - k)$ " in Equation IX represents the convolution of the upsampled image 604 and the interpolating filter,  $f$ , performed 25 at stage 606 in system 600. The filter operation is performed by essentially sliding the lower right pixel of the 2x2 interpolating filter over each pixel of the upsampled image 604. The four pixels of the upsampled image 604 within the 2x2 interpolating filter window are multiplied by the corresponding filter coefficient (i.e., "1" in the illustrated embodiment). The results of the four

multiplications are summed, and the value for the pixel of the upsampled image 604 corresponding to the lower right position of the interpolating filter is replaced by the sum of the four multiplication results. The high resolution data,  $h(n)$ , from the high resolution image 28 is subtracted from the convolution value,  $I_Q(k) f(n - k)$ , to provide an error value. The summation of the squared error over all of the high resolution pixel locations provides a measure of the error to be minimized.

An optimal spatial domain solution can be obtained by taking the derivative of Equation IX with respect to each of the low resolution pixels, and setting it equal to zero as shown in the following Equation X:

10 Equation X

$$\frac{\partial J}{\partial I_Q^*(t)} = 0, \quad t \in \Theta$$

where:

$\Theta$  = the set of quincunx lattice points.

Thus, as can be seen from Equation X, the derivative is taken only at the set of quincunx lattice points, which correspond to the dark pixels in upsampled image 604 in Figure 9. Inserting the equation for  $J$  given in Equation IX into Equation X, and taking the derivative as specified in Equation X, results in the following Equation XI:

Equation XI

$$\sum_k I_Q^*(k) C_{ff}(t - k) = \sum_n h(n) f(n - t), \quad t \in \Theta$$

20 The symbol,  $C_{ff}$ , in Equation XI represents the auto-correlation coefficients of the interpolating filter,  $f$ , as defined by the following Equation XII:

Equation XII

$$C_{ff}(n) = \sum_k f(n) f(n + k)$$

Equation XI can be put into vector form as shown in the following

25 Equation XIII:

Equation XIII

$$\mathbf{C}_{ff} \mathbf{I}_Q^* = \mathbf{h}_f, \quad t \in \Theta$$

where:

$\mathbf{C}_{ff}$  = matrix of auto-correlation coefficients of the interpolating filter,  $f$ .

5  $\mathbf{l}_Q^*$  = vector representing the unknown image data for sub-frame 30H, as well as “don’t care” data (i.e., the image data corresponding to the light pixels in upsampled image 604);

10  $\mathbf{h}_f$  = vector representing a filtered version of the simulated high resolution image 610 using the interpolating filter,  $f$ .

Deleting the rows and columns corresponding to “don’t care” data (i.e., the data that is not in the set of quincunx lattice points,  $\Theta$ ), results in the

15 following Equation XIV:

Equation XIV

$$\tilde{\mathbf{C}}_{ff} \tilde{\mathbf{l}}_Q^* = \tilde{\mathbf{h}}_f$$

where:

$\tilde{\mathbf{l}}_Q^*$  = vector representing only the unknown image data for sub - frame 30H.

20 The above Equation XIV is a sparse non-Toeplitz system representing a sparse system of linear equations. Since the matrix of auto-correlation coefficients is known, and the vector representing the filtered version of the simulated high resolution image 610 is known, Equation XIV can be solved to determine the optimal image data for sub-frame 30H. In one embodiment, sub-frame generation unit 36 is configured to solve Equation XIV to generate sub-frames 30.

VII. Frequency Domain

A frequency domain solution for generating optimal sub-frames 30 according to one embodiment is described in the context of the system 500

shown in Figure 8. Before describing the frequency domain solution, a few properties of the fast fourier transform (FFT) that are applicable to the frequency domain solution are described with reference to Figures 12 and 13.

Figure 12 is a diagram illustrating the effect in the frequency domain of the upsampling of a 4x4 pixel sub-frame 30J according to one embodiment of the present invention. As shown in Figure 12, sub-frame 30J is upsampled by a factor of two by upsampling stage 902 to generate an 8x8 pixel upsampled image 904. The dark pixels in upsampled image 904 represent the sixteen pixels from sub-frame 30J, and the light pixels in upsampled image 904 represent zero values. Taking the FFT of sub-frame 30J results in image ( $L$ ) 906. Taking the FFT of upsampled image 904 results in image ( $L_U$ ) 908. Image ( $L_U$ ) 908 includes four 4x4 pixel portions, which are image portion ( $L_1$ ) 910A, image portion ( $L_2$ ) 910B, image portion ( $L_3$ ) 910C, and image portion ( $L_4$ ) 910D. As shown in Figure 12, image portions 910A-910D are each the same as image 906 (i.e.,  $L_1 = L_2 = L_3 = L_4 = L$ ).

Figure 13 is a diagram illustrating the effect in the frequency domain of the shifting of an 8x8 pixel upsampled sub-frame 904 according to one embodiment of the present invention. As shown in Figure 13, upsampled sub-frame 904 is shifted by shifting stage 1002 to generate shifted image 1004. Taking the FFT of upsampled sub-frame 904 results in image ( $L_U$ ) 1006. Taking the FFT of shifted image 1004 results in image ( $L_{US}$ ) 1008. Image ( $L_{US}$ ) 1008 includes four 4x4 pixel portions, which are image portion ( $LS_1$ ) 1010A, image portion ( $LS_2$ ) 1010B, image portion ( $LS_3$ ) 1010C, and image portion ( $LS_4$ ) 1010D. As shown in Figure 13, image 1008 is the same as image 1006 multiplied by a complex exponential,  $W$ , (i.e.,  $L_{US} = W \cdot L_U$ ), where “•” denotes pointwise multiplication. The values for the complex exponential,  $W$ , are given by the following Equation XV:

Equation XV

$$[W]_{k_1, k_2} = e^{-\frac{j2\pi(k_1+k_2)}{MN}}$$

where:

30  $k_1$  = row coordinate in the FFT domain;

$k_2$  = column coordinate in the FFT domain;

$M$  = number of columns in the image; and

$N$  = number of rows in the image.

The system 500 shown in Figure 8 can be represented mathematically in  
5 an error cost function by the following Equation XVI:

Equation XVI

$$(\mathbf{L}_A^*, \mathbf{L}_B^*) = \underset{(\mathbf{L}_A, \mathbf{L}_B)}{\operatorname{argmin}} J = \underset{(\mathbf{L}_A, \mathbf{L}_B)}{\operatorname{argmin}} \sum_i [\hat{\mathbf{F}}_i (\mathbf{L}_A + \hat{\mathbf{W}}_i \mathbf{L}_B) - \mathbf{H}_i]^H [\hat{\mathbf{F}}_i (\mathbf{L}_A + \hat{\mathbf{W}}_i \mathbf{L}_B) - \mathbf{H}_i]$$

where:

$(\mathbf{L}_A^*, \mathbf{L}_B^*)$  = vectors representing the optimal FFT's of sub-frames 30F and 30G, respectively, shown in Figure 8;

10  $J$  = error cost function to be minimized;

$i$  = index identifying FFT blocks that are averaged (e.g., for image 908 in Figure 12, four blocks are averaged, with  $i=1$  corresponding to block 910A,  $i=2$  corresponding to block 910B,  $i=3$  corresponding to block 910C, and  $i=4$  corresponding to block 910D);

15  $\mathbf{F}$  = matrix representing the FFT of the interpolating filter,  $f$ ;

$\mathbf{L}_A$  = vector representing the FFT of sub-frame 30F shown in Figure 8;

20  $\mathbf{L}_B$  = vector representing the FFT of sub-frame 30G shown in Figure 8;

$\mathbf{W}$  = matrix representing the FFT of the complex coefficient given by Equation XV;

25  $\mathbf{H}$  = vector representing the FFT of the desired high resolution image 28.

The superscript "H" in Equation XVI represents the Hermitian (i.e.,  $\mathbf{X}^H$  is the Hermitian of  $\mathbf{X}$ ). The "hat" over the letters in Equation XVI indicates that those letters represent a diagonal matrix, as defined in the following Equation XVII:

Equation XVII

$$\hat{\mathbf{X}} = \text{diag}(\mathbf{X}) = \begin{pmatrix} X_1 & 0 & 0 & 0 \\ 0 & X_2 & 0 & 0 \\ 0 & 0 & X_3 & 0 \\ 0 & 0 & 0 & X_4 \end{pmatrix}$$

Taking the derivative of Equation XVI with respect to the complex conjugate of  $\mathbf{L}_A$  and setting it equal to zero results in the following Equation 5 XVIII:

Equation XVIII

$$\frac{\partial J}{\partial \overline{\mathbf{L}}_A} = \underbrace{\sum_i \overline{\hat{\mathbf{F}}}_i \hat{\mathbf{F}}_i \mathbf{L}_A}_\mathbf{A} + \underbrace{\sum_i \overline{\hat{\mathbf{F}}}_i \hat{\mathbf{F}}_i \hat{\mathbf{W}}_i \mathbf{L}_B}_\mathbf{B} - \underbrace{\sum_i \overline{\hat{\mathbf{F}}}_i \mathbf{H}_i}_\mathbf{C} = \mathbf{0}$$

Taking the derivative of Equation XVI with respect to the complex conjugate of  $\mathbf{L}_B$  and setting it equal to zero results in the following Equation XIX:

Equation XIX

$$\frac{\partial J}{\partial \overline{\mathbf{L}}_B} = \underbrace{\sum_i \overline{\hat{\mathbf{W}}}_i \overline{\hat{\mathbf{F}}}_i \hat{\mathbf{F}}_i \mathbf{L}_A}_\mathbf{B} + \underbrace{\sum_i \overline{\hat{\mathbf{F}}}_i \hat{\mathbf{F}}_i \mathbf{L}_B}_\mathbf{A} - \underbrace{\sum_i \hat{\mathbf{W}}_i \overline{\hat{\mathbf{F}}}_i \mathbf{H}_i}_\mathbf{D} = \mathbf{0}$$

The horizontal bar over the letters in Equations XVIII and XIX indicates that those letters represent a complex conjugate (i.e.,  $\overline{\mathbf{A}}$  represents the complex conjugate of  $\mathbf{A}$ ).

15 Solving Equations XVIII and XIX for  $\mathbf{L}_A$  and  $\mathbf{L}_B$  results in the following Equations XX and XXI

Equation XX

$$\mathbf{L}_B = (\overline{\hat{\mathbf{B}}} \hat{\mathbf{A}}^{-1} \hat{\mathbf{B}})^{-1} (\mathbf{D} - \hat{\mathbf{A}}^{-1} \mathbf{C})$$

Equation XXI

$$\mathbf{L}_A = \hat{\mathbf{A}}^{-1} (\mathbf{C} - \hat{\mathbf{B}} \mathbf{L}_B)$$

Equations XX and XXI may be implemented in the frequency domain using pseudo-inverse filtering. In one embodiment, sub-frame generation unit 36 is configured to generate sub-frames 30 based on Equations XX and XXI.

## 5 VIII. Adaptive Multi-Pass

An adaptive multi-pass algorithm for generating sub-frames 30 according to one embodiment uses past errors to update estimates for the sub-frame data, and provides fast convergence and low memory requirements. The adaptive multi-pass solution according to one embodiment is described in the context of the system 600 shown in Figure 9. The system 600 shown in Figure 9 can be represented mathematically in an error cost function by the following Equation 10 XXII:

### Equation XXII

$$J^{(n)}(\mathbf{n}) = \left| e^{(n)}(\mathbf{n}) \right|^2 = \left( \sum_{\mathbf{k}} l_Q^{(n)}(\mathbf{k}) f(\mathbf{n} - \mathbf{k}) - h(\mathbf{n}) \right)^2$$

where:

15  $n$  = index identifying the current iteration;

$J^{(n)}(\mathbf{n})$  = error cost function at iteration  $n$ :

$e^{(n)}(n)$  = square root of the error cost function.

$$J^{(n)}(\mathbf{n})$$

**n** and **k** = indices for identifying high resolution pixel locations in images 604 and 610:

20 pixel locations in images 604 and 610;  
 $I_Q^{(n)}(\mathbf{k})$  = image data from upsampled image 604 at  
location  $\mathbf{k}$ :

$f(n - k)$  = filter coefficient of the interpolating filter at a position  $n - k$ ; and

25  $h(n)$  = image data for desired high resolution  
image 28 at location  $n$

As can be seen from Equation XXII, rather than minimizing a global spatial domain error by summing over the entire high resolution image as shown in Equation IX above, a local spatial domain error, which is a function of  $n$ , is being minimized.

A least mean squares (LMS) algorithm is used in one embodiment to determine the update, which is represented in the following Equation XXIII:

Equation XXIII

$$l_{\mathcal{Q}}^{(n+1)}(\mathbf{t}) = l_{\mathcal{Q}}^{(n)}(\mathbf{t}) + \alpha \frac{\partial J^{(n)}(\mathbf{n})}{\partial l_{\mathcal{Q}}^{(n)}(\mathbf{t})}, \quad \mathbf{t} \in \Theta$$

where:

5

$\Theta$  = the set of quincunx lattice points (i.e., the dark pixels in upsampled image 604 in Figure 9); and  
 $\alpha$  = sharpening factor.

Taking the derivative of Equation XXII provides the value for the derivative in Equation XXIII, which is given in the following Equation XXIV:

10

Equation XXIV

$$\frac{\partial J^{(n)}(\mathbf{n})}{\partial l_{\mathcal{Q}}^{(n)}(\mathbf{t})} = 2 \left( \sum_{\mathbf{k}} l_{\mathcal{Q}}^{(n)}(\mathbf{k}) f(\mathbf{n} - \mathbf{k}) - h(\mathbf{n}) \right) f(\mathbf{n} - \mathbf{t})$$

In one embodiment, a block-LMS algorithm using the average gradient over a “region of influence” is used to perform the update, as represented by the following Equation XXV:

15

Equation XXV

$$l_{\mathcal{Q}}^{(n+1)}(\mathbf{t}) = l_{\mathcal{Q}}^{(n)}(\mathbf{t}) + \alpha \sum_{\mathbf{n} \in \Omega} \frac{\partial J^{(n)}(\mathbf{n})}{\partial l_{\mathcal{Q}}^{(n)}(\mathbf{t})}$$

where:

$\Omega$  = region of influence

Figure 14 is a diagram illustrating regions of influence ( $\Omega$ ) 1106 and 1108 for pixels in an upsampled image 1100 according to one embodiment of the

20

present invention. Pixel 1102 of image 1100 corresponds to a pixel for a first sub-frame, and pixel 1104 of image 1100 corresponds to a pixel for a second sub-frame. Region 1106, which includes a 2x2 array of pixels with pixel 1102 in the upper left corner of the 2x2 array, is the region of influence for pixel 1102.

Similarly, region 1108, which includes a 2x2 array of pixels with pixel 1104 in the upper left corner of the 2x2 array, is the region of influence for pixel 1104.

25

Figure 15 is a diagram illustrating the generation of an initial simulated high resolution image 1208 based on an adaptive multi-pass algorithm according to one embodiment of the present invention. An initial set of low resolution sub-frames 30K-1 and 30L-1 are generated based on an original high 5 resolution image 28. In the illustrated embodiment, the initial set of sub-frames 30K-1 and 30L-1 are generated using an embodiment of the nearest neighbor algorithm described above with reference to Figure 5. The sub-frames 30K-1 and 30L-1 are upsampled to generate upsampled image 1202. The upsampled 10 image 1202 is convolved with an interpolating filter 1204, thereby generating a blocked image, which is then multiplied by a factor of 0.5 to generate simulated high resolution image 1208. In the illustrated embodiment, the interpolating filter 1204 is a 2x2 filter with filter coefficients of "1", and with the center of the convolution being the upper left position in the 2x2 matrix. The lower right pixel 1206 of the interpolating filter 1204 is positioned over each pixel in image 1202 15 to determine the blocked value for that pixel position. As shown in Figure 15, the lower right pixel 1206 of the interpolating filter 1204 is positioned over the pixel in the third row and fourth column of image 1202, which has a value of "0". The blocked value for that pixel position is determined by multiplying the filter coefficients by the pixel values within the window of the filter 1204, and adding 20 the results. Out-of-frame values are considered to be "0". For the illustrated embodiment, the blocked value for the pixel in the third row and fourth column of image 1202 is given by the following Equation XXVI

Equation XXVI

$$(1 \times 0) + (1 \times 5) + (1 \times 5) + (1 \times 0) = 10$$

25 The value in Equation XXVI is then multiplied by the factor 0.5, and the result (i.e., 5) is the pixel value for the pixel 1210 in the third row and the fourth column of the initial simulated high resolution image 1208.

30 After the initial simulated high resolution image 1208 is generated, correction data is generated. Figure 16 is a diagram illustrating the generation of correction data based on the adaptive multi-pass algorithm according to one embodiment of the present invention. As shown in Figure 16, the initial simulated high resolution image 1208 is subtracted from the original high

resolution image 28 to generate an error image 1302. Correction sub-frames 1312 and 1314 are generated by averaging 2x2 blocks of pixels in error image 1302. For example, the pixel 1308 in the first column and first row of error image 1302 has a region of influence 1304. The pixel values within the region

5 of influence 1304 are averaged to generate a first correction value (i.e., 0.75). The first correction value is used for the pixel in the first column and the first row of correction sub-frame 1312. Similarly, the pixel 1310 in the second column and second row of error image 1302 has a region of influence 1306. The pixel values within the region of influence 1306 are averaged to generate a second

10 correction value (i.e., 0.75). The second correction value is used for the pixel in the first column and the first row of correction sub-frame 1314.

The correction value in the first row and second column of correction sub-frame 1312 (i.e., 1.38) is generated by essentially sliding the illustrated region of influence box 1304 two columns to the right and averaging those four pixels

15 within the box 1304. The correction value in the second row and first column of correction sub-frame 1312 (i.e., 0.50) is generated by essentially sliding the illustrated region of influence box 1304 two rows down and averaging those four pixels within the box 1304. The correction value in the second row and second column of correction sub-frame 1312 (i.e., 0.75) is generated by essentially

20 sliding the illustrated region of influence box 1304 two columns to the right and two rows down and averaging those four pixels within the box 1304.

The correction value in the first row and second column of correction sub-frame 1314 (i.e., 0.00) is generated by essentially sliding the illustrated region of influence box 1306 two columns to the right and averaging those pixels within

25 the box 1306. Out-of-frame values are considered to be "0". The correction value in the second row and first column of correction sub-frame 1314 (i.e., 0.38) is generated by essentially sliding the illustrated region of influence box 1306 two rows down and averaging those pixels within the box 1306. The correction value in the second row and second column of correction sub-frame

30 1314 (i.e., 0.00) is generated by essentially sliding the illustrated region of influence box 1306 two columns to the right and two rows down and averaging those four pixels within the box 1306.

The correction sub-frames 1312 and 1314 are used to generate updated sub-frames. Figure 17 is a diagram illustrating the generation of updated sub-frames 30K-2 and 30L-2 based on the adaptive multi-pass algorithm according to one embodiment of the present invention. As shown in Figure 17, the

5 updated sub-frame 30K-2 is generated by multiplying the correction sub-frame 1312 by the sharpening factor,  $\alpha$ , and adding the initial sub-frame 30K-1. The updated sub-frame 30L-2 is generated by multiplying the correction sub-frame 1314 by the sharpening factor,  $\alpha$ , and adding the initial sub-frame 30L-1. In the illustrated embodiment, the sharpening factor,  $\alpha$ , is equal to 0.8.

10 In one embodiment, updated sub-frames 30K-2 and 30L-2 are used in the next iteration of the adaptive multi-pass algorithm to generate further updated sub-frames. Any desired number of iterations may be performed. After a number of iterations, the values for the sub-frames generated using the adaptive multi-pass algorithm converge to optimal values. In one embodiment, 15 sub-frame generation unit 36 is configured to generate sub-frames 30 based on the adaptive multi-pass algorithm.

The embodiment of the adaptive multi-pass algorithm described above with reference to Figures 15-17 is for two-position processing. For four-position processing, Equation XXIV becomes the following Equation XXVII:

20 Equation XXVII

$$\frac{\partial J^{(n)}(\mathbf{n})}{\partial l^{(n)}(\mathbf{t})} = 2 \left( \sum_{\mathbf{k}} l^{(n)}(\mathbf{k}) f(\mathbf{n} - \mathbf{k}) - h(\mathbf{n}) \right) f(\mathbf{n} - \mathbf{t})$$

where:

$l^{(n)}$  = low resolution data for the four sub-frames 30;

And Equation XXIII becomes the following Equation XXVIII:

25 Equation XXVIII

$$l^{(n+1)}(\mathbf{t}) = l^{(n)}(\mathbf{t}) + \alpha \frac{\partial J^{(n)}(\mathbf{n})}{\partial l^{(n)}(\mathbf{t})}$$

For four-position processing, there are four sub-frames, so the amount of low resolution data is the same as the amount of high resolution data. Each high resolution grid point contributes one error, and there is no need to average

gradient update as represented in Equation XXV above. Rather, the error at a given location directly gives the update.

As described above, in one embodiment, the adaptive multi-pass algorithm uses a least mean squares (LMS) technique to generate correction data. In another embodiment, the adaptive multi-pass algorithm uses a projection on a convex set (POCS) technique to generate correction data. The adaptive multi-pass solution based on the POCS technique according to one embodiment is described in the context of the system 600 shown in Figure 9. The system 600 shown in Figure 9 can be represented mathematically in an error cost function by the following Equation XXIX:

Equation XXIX

$$|e(\mathbf{n})| = \left| \left( \sum_{\mathbf{k}} l_Q(\mathbf{k}) f(\mathbf{n} - \mathbf{k}) - h(\mathbf{n}) \right) \right|$$

where:

$e(\mathbf{n})$  = error cost function;

$\mathbf{n}$  and  $\mathbf{k}$  = indices identifying high resolution pixel locations;

$l_Q(\mathbf{k})$  = image data from upsampled image 604 at location  $\mathbf{k}$ ;

$f(\mathbf{n} - \mathbf{k})$  = filter coefficient of the interpolating filter at a position  $\mathbf{n} - \mathbf{k}$ ; and

$h(\mathbf{n})$  = image data for desired high resolution image 28 at location  $\mathbf{n}$ .

A constrained set for the POCS technique is defined by the following Equation XXX:

Equation XXX

$$C(\mathbf{n}) = \left\{ l_Q(\mathbf{n}) : \left| \left( \sum_{\mathbf{k}} l_Q(\mathbf{k}) f(\mathbf{n} - \mathbf{k}) - h(\mathbf{n}) \right) \right| \leq \eta \right\}$$

where:

$C(\mathbf{n})$  = constrained set that includes all sub-frame data from upsampled image 604 that is bounded by

parameter,  $\eta$ ; and

$\eta$  = error magnitude bound constraint.

The sub-frame pixel values for the current iteration are determined based on the following Equation XXXI:

5 Equation XXXI

$$l_{\Omega}^{(n+1)}(\mathbf{t}) = \begin{cases} (1 - \lambda) l_{\Omega}^{(n)}(\mathbf{t}) + \lambda \frac{e(\mathbf{n}^*) - \eta}{\|f\|^2} & e(\mathbf{n}^*) > \eta \quad (\mathbf{t} \in \Theta) \\ (1 - \lambda) l_{\Omega}^{(n)}(\mathbf{t}) + \lambda \frac{e(\mathbf{n}^*) + \eta}{\|f\|^2} & e(\mathbf{n}^*) < \eta \\ l_{\Omega}^{(n)}(\mathbf{t}) & e(\mathbf{n}^*) = \eta \end{cases}$$

where:

$n$  = index identifying the current iteration;

$\lambda$  = relaxation parameter; and

10  $\|f\|$  = norm of the coefficients of the interpolating filter.

The symbol,  $\mathbf{n}^*$ , in Equation XXXI represents the location in the region of influence,  $\Omega$ , where the error is a maximum, and is defined by the following Equation XXXII:

15 Equation XXXII

$$\mathbf{n}^* = \operatorname{argmax} \{ \mathbf{n} \in \Omega : |e(\mathbf{n})| \}$$

Figure 18 is a diagram illustrating the generation of correction data based on the adaptive multi-pass algorithm using a POCS technique according to one embodiment of the present invention. In one embodiment, an initial simulated high resolution image 1208 is generated in the same manner as described above with reference to Figure 15, and the initial simulated high resolution image 1208 is subtracted from the original high resolution image 28 to generate an error image 1302. The Equation XXXI above is then used to generate updated sub-frames 30K-3 and 30L-3 from the data in error image 1302. For 20 the illustrated embodiment, it is assumed that relaxation parameter,  $\lambda$ , in

25 the illustrated embodiment, it is assumed that relaxation parameter,  $\lambda$ , in

Equation XXXI is equal to 0.5, and the error magnitude bound constraint,  $\eta$ , is equal to 1.

With the POCS technique, rather than averaging the pixel values within the region of influence to determine a correction value as described above with reference to Figure 16, the maximum error,  $e(n^*)$ , within the region of influence is identified. An updated pixel value is then generated using the appropriate formula from Equation XXXI, which will depend on whether the maximum error,  $e(n^*)$ , within the region of influence is greater than 1, less than 1, or equal to 1 (since  $\eta = 1$  for this example).

For example, the pixel in the first column and first row of error image 1302 has a region of influence 1304. The maximum error within this region of influence 1304 is 1 (i.e.,  $e(n^*) = 1$ ). Referring to Equation XXXI, for the case where  $e(n^*) = 1$ , the updated pixel value is equal to the previous value for this pixel. Referring to Figure 15, the previous value for the pixel in the first column and the first row of sub-frame 30K-1 was 2, so this pixel remains with a value of 2 in updated sub-frame 30K-3. The pixel in the second column and second row of error image 1302 has a region of influence 1306. The maximum error within this region of influence 1306 is 1.5 (i.e.,  $e(n^*) = 1.5$ ). Referring to Equation XXXI, for the case where  $e(n^*) > 1$ , the updated pixel value is equal to half the previous value for this pixel, plus half of the quantity  $(e(n^*) - 1)$ , which is equal to 1.25. Referring to Figure 15, the previous value for the pixel in the first column and the first row of sub-frame 30L-1 was 2, so the updated value for this pixel is 1.25 in updated sub-frame 30L-3.

The region of influence boxes 1302 and 1304 are essentially moved around the error image 1302 in the same manner as described above with reference to Figure 16 to generate the remaining updated values in updated sub-frames 30K-3 and 30L-3 based on Equation XXXI.

## IX. Center Adaptive Multi-Pass

A center adaptive multi-pass algorithm for generating sub-frames 30 according to one embodiment uses past errors to update estimates for sub-frame data and may provide fast convergence and low memory requirements.

The center adaptive multi-pass algorithm modifies the four-position adaptive multi-pass algorithm described above. With the center adaptive multi-pass algorithm, each pixel in each of four sub-frames 30 is centered with respect to a pixel in an original high resolution image 28. The four sub-frames are displayed

5 with display device 26 using four-position processing as described above with reference to Figures 3A-3E.

Figures 19A-19E are schematic diagrams illustrating the display of four sub-frames 1412A, 1422A, 1432A, and 1442A with respect to an original high resolution image 28 according to one embodiment of the present invention. As  
10 shown in Figure 19A, image 28 comprises 8x8 pixels with a pixel 1404 shaded for illustrative purposes.

Figure 19B illustrates the first sub-frame 1412A with respect to image 28. Sub-frame 1412A comprises 4x4 pixels centered on a first set of pixels in image 28. For example, a pixel 1414 in sub-frame 1412A is centered with respect to  
15 pixel 1404 from image 28.

Figure 19C illustrates the second sub-frame 1422A with respect to image 28. Sub-frame 1422A comprises 4x4 pixels centered on a second set of pixels in image 28. For example, a pixel in sub-frame 1422A is centered with respect to a pixel to the right of pixel 1404 from image 28. Two pixels 1424 and 1426 in  
20 sub-frame 1422A overlap pixel 1404 from image 28.

Figure 19D illustrates the third sub-frame 1432A with respect to image 28. Sub-frame 1432A comprises 4x4 pixels centered on a third set of pixels in image 28. For example, a pixel in sub-frame 1432A is centered with respect to a pixel below pixel 1404 from image 28. Pixels 1434 and 1436 in sub-frame  
25 1432A overlap pixel 1404 from image 28.

Figure 19E illustrates the fourth sub-frame 1442A with respect to image 28. Sub-frame 1442A comprises 4x4 pixels centered on a fourth set of pixels in image 28. For example, a pixel in sub-frame 1442A is centered with respect to a pixel diagonally to the right of and below pixel 1404 from image 28. Pixels  
30 1444, 1446, 1448, and 1450 in sub-frame 1442A overlap pixel 1404 from image 28.

When the four sub-frames 1412A, 1422A, 1432A, and 1442A are displayed, nine sub-frame pixels combine to form the displayed representation of each pixel from the original high resolution image 28. For example, nine sub-frame pixels—pixel 1414 from sub-frame 1412A, pixels 1424 and 1426 from 5 sub-frame 1422A, pixels 1434 and 1436 from sub-frame 1432A, and pixels 1444, 1446, 1448, and 1450 from sub-frame 1442A combine to form the displayed representation of pixel 1404 from the original high resolution image 28. These nine sub-frame pixels, however, contribute different amounts of light to the displayed representation of pixel 1404. In particular, pixels 1424, 1426, 10 1434, and 1436 from sub-frames 1422A and 1432A, respectively, each contribute approximately one-half as much light as pixel 1414 from sub-frame 1412A as illustrated by only a portion of pixels 1424, 1426, 1434, and 1436 overlapping pixel 1404 in Figures 19C and 19D. Similarly, pixels 1444, 1446, 1448, and 1450 from sub-frame 1442A each contribute approximately one- 15 fourth as much light as pixel 1414 from sub-frame 1412A as illustrated by only a portion of pixels 1444, 1446, 1448, and 1450 overlapping pixel 1404 in Figures 19C and 19D.

Sub-frame generation unit 36 generates the initial four sub-frames 1412A, 1422A, 1432A, and 1442A from the high resolution image 28. In one 20 embodiment, sub-frames 1412A, 1422A, 1432A, and 1442A may be generated using an embodiment of the nearest neighbor algorithm described above with reference to Figure 5. In other embodiments, sub-frames 1412A, 1422A, 1432A, and 1442A may be generated using other algorithms. For error processing, the sub-frames 1412A, 1422A, 1432A, and 1442A are upsampled 25 to generate an upsampled image, shown as sub-frame 30M in Figure 20.

Figure 20 is a block diagram illustrating a system 1500 for generating a simulated high resolution image 1504 for four-position processing based on sub-frame 30M using a center adaptive multi-pass algorithm according to one embodiment of the present invention. In the embodiment illustrated in Figure 30 20, sub-frame 30M is an 8x8 array of pixels. Sub-frame 30M includes pixel data for four 4x4 pixel sub-frames for four-position processing. Pixels A1-A16 represent pixels from sub-frame 1412A, pixels B1-B16 represent pixels from

sub-frame 1422A, pixels C1-C16 represent pixels from sub-frame 1432A, and pixels D1-D16 represent pixels from sub-frame 1442A.

The sub-frame 30M is convolved with an interpolating filter at convolution stage 1502, thereby generating the simulated high resolution image 1504. In

5 the illustrated embodiment, the interpolating filter is a 3x3 filter with the center of the convolution being the center position in the 3x3 matrix. The filter coefficients of the first row are "1/16", "2/16", "1/16", the filter coefficients of the second row are "2/16", "4/16", "2/16", and the filter coefficients of the last row are "1/16", "2/16", "1/16".

10 The filter coefficients represent the relative proportions that nine sub-frame pixels make toward the displayed representation of a pixel of the high resolution image 28. Recalling the example of Figure 19 above, pixels 1424, 1426, 1434, and 1436 from sub-frames 1422A and 1432A, respectively, each contribute approximately one-half as much light as pixel 1414 from sub-frame 15 1412A, and pixels 1444, 1446, 1448, and 1450 from sub-frame 1442A each contribute approximately one-fourth as much light as pixel 1414 from sub-frame 1412A. The values of the sub-frame pixels 1414, 1424, 1426, 1434, 1436, 1444, 1446, 1448, and 1450 correspond to the A6, B5, B6, C2, C6, D1, D5, D2, and D6 pixels in sub-frame image 30M, respectively. Thus, the pixel A6<sub>SIM</sub> for 20 the simulated image 1504 (which corresponds to pixel 1404 in Figure 19) is calculated from the values in the sub-frame image 30M as follows in Equation XXXIII:

Equation XXXIII

$$25 \quad A6_{SIM} = ((1 \times D1) + (2 \times C2) + (1 \times D2) + (2 \times B5) + (4 \times A6) + (2 \times B6) \\ + (1 \times D5) + (2 \times C6) + (1 \times D6)) / 16$$

The image data is divided by a factor of 16 to compensate for the relative proportions that the nine sub-frame pixels contribute to each displayed pixel.

30 After the simulated high resolution image 1504 is generated, correction data is generated. Figure 21 is a block diagram illustrating the generation of correction data using a center adaptive multi-pass algorithm in a system 1520

according to one embodiment of the present invention. The simulated high resolution image 1504 is subtracted on a pixel-by-pixel basis from high resolution image 28 at subtraction stage 1522. In one embodiment, the resulting error image data is filtered by an error filter 1526 to generate an error 5 image 1530. In the illustrated embodiment, the error filter is a 3x3 filter with the center of the convolution being the center position in the 3x3 matrix. The filter coefficients of the first row are "1/16", "2/16", "1/16", the filter coefficients of the second row are "2/16", "4/16", "2/16", and the filter coefficients of the last row are "1/16", "2/16", "1/16". The filter coefficients represent the proportionate 10 differences between a low resolution sub-frame pixel and the nine pixels of the high resolution image 28. As illustrated in Figures 19B, the error value in error image 1530 for low resolution sub-frame pixel 1414 is measured against pixel 1404 of the high resolution image 28 and the eight high resolution pixels immediately adjacent to pixel 1404. With the above filter coefficients, the high 15 resolution pixels above, below, to the left, and the right of pixel 1404 are weighted twice as much as the high resolution pixels adjacent to the corners of pixel 1404 in calculating the error value corresponding to pixel 1414. Similarly, pixel 1404 is weighted twice as much as the four high resolution pixels the high resolution pixels above, below, to the left, and the right of pixel 1404 in 20 calculating the error value corresponding to pixel 1414.

Four correction sub-frames (not shown) associated with the initial sub-frames 1412A, 1422A, 1432A, and 1442A, respectively, are generated from the error image 1530. Four updated sub-frames 1412B, 1422B, 1432B, and 1442B are generated by multiplying the correction sub-frames by the sharpening factor, 25  $\alpha$ , and adding the initial sub-frames 1412A, 1422A, 1432A, and 1442A, respectively. The sharpening factor,  $\alpha$ , may be different for different iterations of the center adaptive multi-pass algorithm. In one embodiment, the sharpening factor,  $\alpha$ , may decrease between successive iterations. For example, the sharpening factor,  $\alpha$ , may be "3" for a first iteration, "1.8" for a second iteration, 30 and "0.5" for a third iteration.

In one embodiment, updated sub-frames 1412B, 1422B, 1432B, and 1442B are used in the next iteration of the center adaptive multi-pass algorithm

to generate further updated sub-frames. Any desired number of iterations may be performed. After a number of iterations, the values for the sub-frames generated using the center adaptive multi-pass algorithm converge to optimal values. In one embodiment, sub-frame generation unit 36 is configured to

5 generate sub-frames 30 based on the center adaptive multi-pass algorithm.

In the embodiment of the center adaptive multi-pass algorithm described above, the numerator and denominator values of the filter coefficient were selected to be powers of 2. By using powers of 2, processing in digital systems may be expedited. In other embodiments of the center adaptive multi-pass

10 algorithm, other filter coefficient values may be used.

In other embodiments, the center adaptive multi-pass algorithm just described may be modified to generate two sub-frames for two-position processing. The two sub-frames are displayed with display device 26 using two-position processing as described above with reference to Figures 2A-2C.

15 With two-position processing, pixels B1-B16 and C1-C16 in image 30M (shown in Figure 20) are zero, and the interpolating filter comprises a 3x3 array with the first row of values being "1/8", "2/8", "1/8", the second row of values being "2/8", "4/8", "2/8", and the third row of values being "1/8", "2/8", "1/8". The error filter for two-position processing is the same as the error filter for four-position  
20 processing.

In other embodiments, the center adaptive multi-pass algorithm may be performed in one pass for any number of iterations by merging the calculations of each iteration into a single step for each sub-frame pixel value. In this way, each sub-frame pixel value is generated without explicitly generating simulation, error, and correction sub-frames for each iteration. Rather, each sub-frame  
25 pixel value is independently calculated from intermediate values which are calculated from the original image pixel values.

#### X. Simplified Center Adaptive Multi-Pass

30 A simplified center adaptive multi-pass algorithm for generating sub-frames 30 according to one embodiment uses past errors to update estimates for sub-frame data and provides fast convergence and low memory

requirements. The simplified center adaptive multi-pass algorithm modifies the four-position adaptive multi-pass algorithm described above. With the simplified center adaptive multi-pass algorithm, each pixel in each of four sub-frames 30 is centered with respect to a pixel in an original high resolution image 28 as

5 described above with reference to Figures 19A-19E. The four sub-frames are displayed with display device 26 using four-position processing as described above with reference to Figures 3A-3E.

Referring to Figures 19A-19E, sub-frame generation unit 36 generates the initial four sub-frames 1412A, 1422A, 1432A, and 1442A from the high

10 resolution image 28. In one embodiment, sub-frames 1412A, 1422A, 1432A, and 1442A may be generated using an embodiment of the nearest neighbor algorithm described above with reference to Figure 5. In other embodiments, sub-frames 1412A, 1422A, 1432A, and 1442A may be generated using other algorithms. For error processing, the sub-frames 1412A, 1422A, 1432A, and 15 1442A are upsampled to generate an upsampled image, shown as sub-frame 30M in Figure 22.

Figure 22 is a block diagram illustrating a system 1600 for generating a simulated high resolution image 1604 for four-position processing based on sub-frame 30N using a simplified center adaptive multi-pass algorithm according

20 to one embodiment of the present invention. In the embodiment illustrated in Figure 22, sub-frame 30N is an 8x8 array of pixels. Sub-frame 30N includes pixel data for four 4x4 pixel sub-frames for four-position processing. Pixels A1-A16 represent pixels from sub-frame 1412A, pixels B1-B16 represent pixels from sub-frame 1422A, pixels C1-C16 represent pixels from sub-frame 1432A, 25 and pixels D1-D16 represent pixels from sub-frame 1442A.

The sub-frame 30N is convolved with an interpolating filter at convolution stage 1602, thereby generating the simulated high resolution image 1604. In the illustrated embodiment, the interpolating filter is a 3x3 filter with the center of the convolution being the center position in the 3x3 matrix. The filter coefficients 30 of the first row are "0", "1/8", "0", the filter coefficients of the second row are "1/8", "4/8", "1/8", and the filter coefficients of the last row are "0", "1/8", "0".

The filter coefficients approximate the relative proportions that five sub-frame pixels make toward the displayed representation of a pixel of the high resolution image 28. Recalling the example of Figure 19 above, pixels 1424, 1426, 1434, and 1436 from sub-frames 1422A and 1432A, respectively, each 5 contribute approximately one-half as much light as pixel 1414 from sub-frame 1412A, and pixels 1444, 1446, 1448, and 1450 from sub-frame 1442A each contribute approximately one-fourth as much light as pixel 1414 from sub-frame 1412A. With the simplified center adaptive multi-pass algorithm, the 10 contributions from pixels 1444, 1446, 1448, and 1450, referred to as the “corner pixels”, are ignored in calculating the pixel value for pixel 1414 as indicated by the filter coefficients of 0 associated with the corner pixels.

The values of the sub-frame pixels 1414, 1424, 1426, 1434, 1436, 1444, 1446, 1448, and 1450 correspond to the A6, B5, B6, C2, C6, D1, D5, D2, and 15 D6 pixels in sub-frame image 30N, respectively. Thus, the pixel  $A_{6SIM}$  for the simulated image 1504 (which corresponds to pixel 1404 in Figure 19) is calculated from the values in the sub-frame image 30N as follows in Equation XXXIV:

Equation XXXIV

$$20 \quad A_{6SIM} = ((0 \times D1) + (1 \times C2) + (0 \times D2) + (1 \times B5) + (4 \times A6) + (1 \times B6) \\ + (0 \times D5) + (1 \times C6) + (0 \times D6)) / 8$$

Equation XXXIV simplifies to Equation XXXV:

Equation XXXV

$$25 \quad A_{6SIM} = (C2 + B5 + (4 \times A6) + B6 + C6) / 8$$

The image data is divided by a factor of 8 to compensate for the relative proportions that the five sub-frame pixels contribute to each displayed pixel.

30 After the simulated high resolution image 1604 is generated, correction data is generated. Figure 23 is a block diagram illustrating the generation of correction data using a center adaptive multi-pass algorithm in a system 1700

according to one embodiment of the present invention. The simulated high resolution image 1604 is subtracted on a pixel-by-pixel basis from high resolution image 28 at subtraction stage 1702 to generate an error image 1704.

Four correction sub-frames (not shown) associated with the initial sub-frames 1412A, 1422A, 1432A, and 1442A, respectively, are generated from the error image 1704. Four updated sub-frames 1704A, 1704B, 1704C, and 1704D are generated by multiplying the correction sub-frames by the sharpening factor,  $\alpha$ , and adding the initial sub-frames 1412A, 1422A, 1432A, and 1442A, respectively. The sharpening factor,  $\alpha$ , may be different for different iterations of the simplified center adaptive multi-pass algorithm. In one embodiment, the sharpening factor,  $\alpha$ , may decrease between successive iterations. For example, the sharpening factor,  $\alpha$ , may be "3" for a first iteration, "1.8" for a second iteration, and "0.5" for a third iteration.

In one embodiment, updated sub-frames 1704A, 1704B, 1704C, and 1704D are used in the next iteration of the simplified center adaptive multi-pass algorithm to generate further updated sub-frames. Any desired number of iterations may be performed. After a number of iterations, the values for the sub-frames generated using the simplified center adaptive multi-pass algorithm converge to optimal values. In one embodiment, sub-frame generation unit 36 is configured to generate sub-frames 30 based on the center adaptive multi-pass algorithm.

In the embodiment of the simplified center adaptive multi-pass algorithm described above, the numerator and denominator values of the filter coefficient were selected to be powers of 2. By using powers of 2, processing in digital systems may be expedited. In other embodiments of the simplified center adaptive multi-pass algorithm, other filter coefficient values may be used.

In other embodiments, the simplified center adaptive multi-pass algorithm may be performed in one pass for any number of iterations by merging the calculations of each iteration into a single step for each sub-frame pixel value. In this way, each sub-frame pixel value is generated without explicitly generating simulation, error, and correction sub-frames for each iteration. Rather, each

sub-frame pixel value is independently calculated from intermediate values which are calculated from the original image pixel values.

#### XI. Adaptive Multi-Pass with History

5 An adaptive multi-pass algorithm with history for generating sub-frames  
30 according to one embodiment uses past errors to update estimates for sub-  
frame data and may provide fast convergence and low memory requirements.  
The adaptive multi-pass algorithm with history modifies the four-position  
adaptive multi-pass algorithm described above by using history values to  
10 generate sub-frames in one pass of the algorithm. The four sub-frames are  
displayed with display device 26 using four-position processing as described  
above with reference to Figures 3A-3E.

At least two methods for implementing the adaptive multi-pass algorithm  
may be used. First, the adaptive multi-pass algorithm may be performed in  
15 multiple iterations as described above for the adaptive multi-pass, the center  
adaptive multi-pass, and the simplified center adaptive multi-pass algorithms.  
With multiple iterations, (1) initial sub-frames are generated, (2) a simulated  
image is generated, (3) correction data is calculated by comparing the simulated  
image with the original image, and (4) updated sub-frames are generated using  
20 the correction data. Steps (2) through (4) are then repeated for each iteration.

The adaptive multi-pass algorithm may also be implemented by  
calculating each final sub-frame pixel value in one pass using a region of  
influence for each final sub-frame pixel value. With this method, the size of the  
region of influence corresponds to the number of iterations to be performed as  
25 shown in Figures 24A-24C. As will be described below, the region of influence  
may be simplified as shown in Figures 27 and 31.

Figures 24A-24C are block diagrams illustrating regions of influence for a  
pixel 1802 for different numbers of iterations of the adaptive multi-pass  
algorithm. Figure 24A illustrates a region of influence 1804 for pixel 1802 in an  
30 image 1800 for one iteration of the adaptive multi-pass algorithm. As shown in  
Figure 24A, the region of influence 1804 comprises a 4x4 array of pixels with  
pixel 1802 centered in the region of influence 1804 as shown. The region of

influence 1804 encompasses the pixel values used to generate the initial, simulation, and correction values for pixel 1802 using one iteration of the adaptive multi-pass algorithm.

For two iterations of the adaptive multi-pass algorithm, a region of

5 influence 1806 expands to a 6x6 array with pixel 1802 centered in the region of influence 1806 as shown in Figure 24B. The region of influence 1806 for pixel 1802 comprises a 6x6 array of pixels and encompasses the pixel values used to generate the initial, simulation, and correction values for pixel 1802 using two iterations of the adaptive multi-pass algorithm.

10 As shown in Figure 24C, a region of influence 1808 further expands to an 8x8 array for three iterations of the adaptive multi-pass algorithm. The region of influence 1808 for pixel 1802 comprises an 8x8 array of pixels with pixel 1802 centered in the region of influence 1808 as shown and encompasses the pixel values used to generate the initial, simulation, and correction values for pixel

15 1802 using three iterations of the adaptive multi-pass algorithm. In particular, the region of influence 1808 covers eight rows of the image 1800.

The size of a region of influence may be generalized by noting that for  $n$  iterations, the region of influence comprises a  $(2n + 2) \times (2n + 2)$  array.

With the one pass method for implementing the adaptive multi-pass

20 algorithm, each final sub-frame pixel value is calculated by shifting the region of influence with respect to the pixel value corresponding to the final sub-frame pixel value. Figure 25 is a block diagram illustrating a region of influence 1904 of a pixel 1902 with respect to an image 1900 for three iterations of the adaptive multi-pass algorithm. In Figure 25, the final sub-frame pixel value

25 corresponding to pixel 1902 is calculated using the pixel values encompassed by the region of influence 1904. To calculate the final sub-frame pixel value corresponding to a pixel 1906, the region of influence 1904 is shifted by one pixel to the right (not shown) as indicated by an arrow 1908. Similarly, the region of influence 1904 is shifted down by one pixel (not shown) as indicated

30 by an arrow 1912 to calculate the final sub-frame pixel value corresponding to a pixel 1910.

In one embodiment, the final sub-frame pixel values of image 1900 may be calculated in a raster pattern where the values are calculated row-by-row from left to right beginning with the top row and finishing with the bottom row. In other embodiments, the final sub-frame pixel values may be calculated 5 according to other patterns or in other orders.

Figure 26 is a block diagram illustrating calculated history values in a region of influence 2004 of a pixel 2002 for three iterations of the adaptive multi-pass algorithm where the final sub-frame pixel values are calculated in a raster pattern. The shaded pixels in the region of influence 2004 comprise history 10 values, i.e., final sub-frame pixel values calculated prior to calculating the final sub-frame pixel value for pixel 2002. With a raster pattern, the final sub-frame pixel values are calculated for each row above pixel 2002 and for each pixel on the same row and to the left of pixel 2002.

By using history values and ignoring the last row of initial values, the 15 region of influence 2004 shown in Figure 26 for pixel 2002 may be simplified. Figure 27 is a block diagram illustrating calculated history values in a simplified region of influence 2006 of pixel 2002 for three iterations of the adaptive multi-pass algorithm with history where the final sub-frame pixel values are calculated in a raster pattern. The simplified region of influence 2006 comprises five 20 rows—one row 2008 of history values, one row 2010 of both history values and initial values, and three rows 2012 of initial values. The simplified region of influence 2006 does not include the first two rows of history values and the last row of initial values from the region of influence 2004.

The initial history values 2008 may be set to be equal to the 25 corresponding pixel values from the first row of the original image or may be set to zero. The initial values in rows 2010 and 2012 may be set to zero initially or may be set to be equal to the calculated initial values from a column 2016. The final sub-frame pixel value corresponding to pixel 2002 may be calculated using the history and initial values from the simplified region of influence 2006 using 30 the following algorithm.

First, the initial pixel values for the pixels in column 2016 of the region of influence 2006 are calculated using the original image pixel values. In one

embodiment, the initial pixel values are calculated by averaging each pixel value with three other pixel values. Other algorithms may be used in other embodiments. Next, the simulated pixel values for the pixels in column 2016

are calculated by convolving the initial pixel values with a simulation kernel. The

5 simulation kernel comprises a 3x3 array with the first row of values being "1/4", "1/4", and "0", the second row of values being "1/4", "1/4", and "0", and the third row of values being "0", "0", and "0". Error values are generated for each pixel in column 2016 by subtracting the simulated pixel values from the original image pixel values.

10 After the error values of column 2016 are calculated, the simulated pixel values for the pixels in a column 2018 are calculated by convolving the initial pixel values with the simulation kernel. Error values are generated for each pixel in column 2018 by subtracting the simulated pixel values from the original image pixel values. The correction values for the pixels in column 2018 are

15 calculated by convolving the error values with an error kernel. The error kernel comprises a 3x3 array with the first row of values being "0", "0", and "0", the second row of values being "0", "1/4", and "1/4", and the third row of values being "0", "1/4", and "1/4". The adaptive pixel values for the pixels in column 2018 are calculated by multiplying the correction values by a sharpening factor,

20  $\alpha$ , and adding the product to the initial values.

After the adaptive pixel values of column 2018 are calculated, the simulated pixel values for the pixels in a column 2020 are calculated by convolving the initial pixel values with the simulation kernel. Error values are generated for each pixel in column 2020 by subtracting the simulated pixel

25 values from the original image pixel values. The correction values for the pixels in column 2020 are calculated by convolving the error values with the error kernel. The adaptive pixel values for the pixels in column 2020 are calculated by multiplying the correction values by a sharpening factor,  $\alpha$ , and adding the product to the initial values.

30 After the adaptive pixel values of column 2020 are calculated, the simulated pixel values for the pixels in a column 2022 are calculated by convolving the initial pixel values with the simulation kernel. Error values are

generated for each pixel in column 2022 by subtracting the simulated pixel values from the original image pixel values. The correction values for the pixels in column 2022 are calculated by convolving the error values with the error kernel.

5 The final sub-frame pixel value corresponding to pixel 2002 is calculated using the values generated by the above algorithm, the history values, and a sharpening factor,  $\alpha$ .

Intermediate calculations used in calculating the final sub-frame pixel value corresponding to a given pixel may be re-used in calculating the final sub-frame pixel value corresponding to a pixel adjacent to the given pixel. For example, intermediate calculations used in calculating the final sub-frame pixel value for pixel 2002 may be re-used in calculating the final sub-frame pixel value of a pixel to the right of pixel 2002. As a result, certain redundant calculations may be omitted.

15 The sharpening factor,  $\alpha$ , in the above algorithm may be different in calculating values of different columns using the adaptive multi-pass algorithm with history. For example, the sharpening factor,  $\alpha$ , may be "3" for calculating the adaptive pixel values of column 2018, "1.8" for calculating the adaptive pixel values of column 2020, and "0.5" for calculating the final sub-frame pixel value 20 corresponding to pixel 2002.

Although the above algorithm was described for three iterations of the adaptive multi-pass algorithm, the algorithm can be expanded or reduced to apply to any number of iterations by increasing or decreasing the number of columns and/or the number of pixels in each column used in the above 25 algorithm in accordance with the region of influence for the number of iterations.

Figure 28 is a block diagram illustrating the simplified region of influence 2006 of pixel 2002 with respect to image 1900 for three iterations of the adaptive multi-pass algorithm with history where the final sub-frame pixel values are calculated in a raster pattern. In Figure 28, the final sub-frame pixel value 30 corresponding to pixel 2002 is calculated using the pixel values encompassed by the region of influence 2006 as just described in the above algorithm. To calculate the final sub-frame pixel value corresponding to a pixel 2028, the

region of influence 2006 is shifted by one pixel to the right (not shown) as indicated by the arrow 1908. Similarly, the region of influence 2006 is shifted down by one pixel (not shown) as indicated by the arrow 1912 to calculate the final sub-frame pixel value corresponding to a pixel 2030.

5       Figure 29 is a block diagram illustrating portions of sub-frame generation unit 36 according to one embodiment. In this embodiment, sub-frame generation unit 36 comprises a processor 2100, a main memory 2102, a controller 2104, and a memory 2106. Controller 2104 is coupled to processor 2100, main memory 2102, and a memory 2106. Memory 2106 comprises a 10 relatively large memory that includes an original image 28 and a sub-frame image 30P. Main memory 2102 comprises a relatively fast memory that includes a sub-frame generation module 2110, temporary variables 2112, original image rows 28A from original image 28, and a sub-frame image row 30P-1 from sub-frame image 30P.

15      Processor 2100 accesses instructions and data from main memory 2102 and memory 2106 using controller 2104. Processor 2100 executes instructions and stores data in main memory 2102 and memory 2106 using controller 2104.

20      Sub-frame generation module 2110 comprises instructions that are executable by processor 2100 to implement the adaptive multi-pass algorithm with history. In response to being executed by processor 2100, sub-frame generation module 2110 causes a set of original image rows 28A and a sub-frame image row 30P-1 to be copied into main memory 2102. Sub-frame generation module 2110 causes the final sub-frame pixel values to be generated for each row using the pixel values in original image rows 28A and 25 sub-frame image row 30P-1 according to the adaptive multi-pass algorithm with history. In generating the final sub-frame pixel values, sub-frame generation module 2110 causes temporary values to be stored as temporary variables 2112. After generating the final sub-frame pixel values for a sub-frame image row, sub-frame generation module 2110 causes the row to be stored as sub-frame image 30P and causes a next row of pixel values to be read from original 30 image 28 and stored in original image rows 28A.

5 In one embodiment where sub-frame generation module 2110 implements three iterations of the adaptive multi-pass algorithm with history, original image rows 28A comprises four rows of original image 28. In other embodiments, original image rows 28A comprises other numbers of rows of original image 28.

In one embodiment, sub-frame generation unit 36 generates four sub-frames from sub-frame image 30P. The four sub-frames are displayed with display device 26 using four-position processing as described above with reference to Figures 3A-3E.

10 In other embodiments, sub-frame generation unit 36 comprises an application-specific integrated circuit (ASIC) that incorporates the functions of the components shown in Figure 29 into an integrated circuit. In these embodiments, main memory 2102 may be included in the ASIC and memory 2106 may be included in or external to the ASIC. The ASIC may comprise any 15 combination of hardware and software or firmware components.

In other embodiments, the adaptive multi-pass algorithm with history may be used to generate two sub-frames for two-position processing. The two sub-frames are displayed with display device 26 using two-position processing as described above with reference to Figures 2A-2C. With two-position 20 processing, the simulation kernel comprises a 3x3 array with the first row of values being "1/2", "1/2", "0", the second row of values being "1/2", "1/2", "0", and the third row of values being "0", "0", "0".

## XII. Simplified Center Adaptive Multi-Pass with History

25 A simplified center adaptive multi-pass algorithm with history for generating sub-frames 30 according to one embodiment uses past errors to update estimates for sub-frame data and may provide fast convergence and low memory requirements. The simplified center adaptive multi-pass algorithm with history modifies the adaptive multi-pass algorithm with history by changing the 30 values in the simulation kernel and omitting the error kernel in generating four sub-frames in one pass of the algorithm. The four sub-frames are displayed

with display device 26 using four-position processing as described above with reference to Figures 3A-3E.

With reference to Figure 27, the initial history values 2008 may be set to be equal to the corresponding pixel values from the first row of the original

5 image or may be set to zero. The initial values in rows 2010 and 2012 may be set to zero initially or may be set to be equal to the calculated initial values from column 2016. The final sub-frame pixel value corresponding to pixel 2002 may be calculated using the history and initial values from the simplified region of influence 2006 using the following algorithm. The simplified center adaptive

10 multi-pass algorithm with history may be implemented as follows.

First, the initial pixel values for the pixels in column 2016 are calculated. The initial pixel values may be calculated using the nearest neighbor algorithm or any other suitable algorithm.

After the initial pixel values of column 2016 are calculated, the simulated 15 pixel values for the pixels in a column 2018 are calculated by convolving the initial pixel values with a simulation kernel. The simulation kernel comprises a 3x3 array with the first row of values being "0", "1/8", and "0", the second row of values being "1/8", "4/8", and "1/8", and the third row of values being "0", "1/8", and "0". The correction values for the pixels in column 2018 are calculated by 20 subtracting the simulated pixel values from the original image pixel values. The adaptive pixel values for the pixels in column 2018 are calculated by multiplying the correction values by a sharpening factor,  $\alpha$ , and adding the product to the initial values.

After the simulation values of column 2018 are calculated, the simulated 25 pixel values for the pixels in a column 2020 are calculated by convolving the initial pixel values with the simulation kernel. The correction values for the pixels in column 2020 are calculated by subtracting the simulated pixel values from the original image pixel values. The adaptive pixel values for the pixels in column 2020 are calculated by multiplying the correction values by a sharpening 30 factor,  $\alpha$ , and adding the product to the initial values.

After the simulation values of column 2020 are calculated, the simulated pixel values for the pixels in a column 2022 are calculated by convolving the

initial pixel values with the simulation kernel. The correction values for the pixels in column 2022 are calculated by subtracting the simulated pixel values from the original image pixel values.

5 The final sub-frame pixel value corresponding to pixel 2002 is calculated using the values generated by the above algorithm, the history values, and a sharpening factor,  $\alpha$ .

10 Intermediate calculations used in calculating the final sub-frame pixel value corresponding to a given pixel may be re-used in calculating the final sub-frame pixel value corresponding to a pixel adjacent to the given pixel. For example, intermediate calculations used in calculating the final sub-frame pixel value for pixel 2002 may be re-used in calculating the final sub-frame pixel value of a pixel to the right of pixel 2002. As a result, certain redundant calculations may be omitted.

15 The sharpening factor,  $\alpha$ , in the above algorithm may be different in calculating values of different columns using the simplified center adaptive multi-pass algorithm with history. For example, the sharpening factor,  $\alpha$ , may be "3" for calculating the adaptive pixel values of column 2018, "1.8" for calculating the adaptive pixel values of column 2020, and "0.5" for calculating the final sub-frame pixel value corresponding to pixel 2002.

20 Although the above algorithm was described for three iterations of the simplified center adaptive multi-pass algorithm, the algorithm can be expanded or reduced to apply to any number of iterations by increasing or decreasing the number of columns and the number of pixels in each column used in the above algorithm in accordance with the region of influence for the number of iterations.

25 In one embodiment of sub-frame generation unit 36 (shown in Figure 29), sub-frame generation module 2110 implements the simplified center adaptive multi-pass algorithm with history. In another embodiment, sub-frame generation unit 36 comprises an ASIC that implements the simplified center adaptive multi-pass algorithm with history.

### XIII. Center Adaptive Kernel with History

A center adaptive multi-pass algorithm with history for generating sub-frames 30 according to one embodiment uses past errors to update estimates for sub-frame data and may provide fast convergence and low memory

5 requirements. The center adaptive multi-pass algorithm with history generates two sub-frames in one pass of the algorithm and modifies the adaptive multi-pass algorithm with history by changing simulation and error kernels. The center adaptive multi-pass algorithm with history also generates error values associated with the row of history values used in the simplified region of  
10 influence and stores these values along with the row of history values. The two sub-frames are displayed with display device 26 using two-position processing as described above with reference to Figures 2A-2C.

With two-position processing, the two sub-frames may be intertwined into a single sub-frame image 2200 as illustrated in Figure 30. Within the image  
15 2200, a set of pixels 2202, illustrated with a first type of shading, comprises the first sub-frame, and a set of pixels 2204, illustrated with a second type of shading, comprises the second sub-frame. The remaining set of un-shaded pixels 2206 comprise zero values that represent unused sub-frames.

Figure 31 is a block diagram illustrating pixels 2202, pixels 2204, history  
20 values 2222 (illustrated with a third type of shading), and error values 2224 (illustrated with a fourth type of shading) in a simplified region of influence 2210 of a pixel 2212 for three iterations of the center adaptive multi-pass algorithm with history where the final sub-frame pixel values are calculated in a raster pattern. The simplified region of influence 2210 comprises five rows—one row  
25 2214 of history values and error values, one row 2216 of history values and initial values, and three rows 2218 of initial values. The simplified region of influence 2210 does not include the two rows of history and error values above row 2214 and the row of initial values below rows 2218.

The error values 2224 are each calculated using Equation XXXVI.  
30 Equation XXXVI

$$\text{error} = ((1 \times \text{error}_{\text{left\_pixel}}) + (2 \times \text{error})) + (1 \times \text{error}_{\text{right\_pixel}})) / 4$$

Because the error values 2224 are signed values that may contain more bits than a pixel value, the error values 2224 calculated using Equation XXXVI are adjusted using a mapping or a look-up table before being stored in row 2214 as shown in Figure 31. The following pseudo code may be used to map the error values 2224 according to one embodiment.

```
10      temp = error_left+2*error+error_right; // 1x 2x 1x
      temp = temp/4; // divide by 4
      if( temp < -127 ) temp = -127; // clip value
      if( temp > 127 ) temp = 127; // clip value
      temp += 127; // shift to make non-zero
```

The initial history values 2222 may be set to be equal to the corresponding pixel values from the first row of the original image or may be set to zero. The initial error values 2224 may be set to zero. The initial values in rows 2216 and 2218 may be set to zero initially or may be set to be equal to the calculated initial values from a column 2226. The final sub-frame pixel value corresponding to pixel 2212 may be calculated using the history, error, and initial values from the simplified region of influence 2210 using the following algorithm.

20 First, the initial pixel values for the pixels in column 2226 of the region of influence 2210 are calculated using the original image pixel values. In one embodiment, the initial pixel values are calculated using the nearest neighbor algorithm. Other algorithms may be used in other embodiments. Next, the simulated pixel values for the pixels in column 2226 are calculated by  
25 convolving the initial pixel values with one of two simulation kernels. The first simulation kernel is used when pixel 2212 comprises a non-zero value and comprises a 3x3 array with the first row of values being "1/8", "0", and "1/8", the second row of values being "0", "4/8", and "0", and the third row of values being "1/8", "0", and "1/8". The second simulation kernel is used when pixel 2212  
30 comprises a zero value and comprises a 3x3 array with the first row of values being "0", "2/8", and "0", the second row of values being "2/8", "0", and "2/8", and the third row of values being "0", "2/8", and "0". Error values are generated for each pixel in column 2226 by subtracting the simulated pixel values from the original image pixel values.

After the error values of column 2226 are calculated, the simulated pixel values for the pixels in a column 2228 are calculated by convolving the initial pixel values with the appropriate simulation kernel. Error values are generated for each pixel in column 2228 by subtracting the simulated pixel values from the original image pixel values. The correction values for the pixels in column 2228 are calculated by convolving the error values with an error kernel. The error kernel comprises a 3x3 array with the first row of values being "1/16", "2/16", and "1/16", the second row of values being "2/16", "4/16", and "2/16", and the third row of values being "1/16", "2/16", and "1/16". The adaptive pixel values for the pixels in column 2228 are calculated by multiplying the correction values by a sharpening factor,  $\alpha$ , and adding the product to the initial values.

After the adaptive pixel values of column 2228 are calculated, the simulated pixel values for the pixels in a column 2230 are calculated by convolving the initial pixel values with the appropriate simulation kernel. Error values are generated for each pixel in column 2230 by subtracting the simulated pixel values from the original image pixel values. The correction values for the pixels in column 2230 are calculated by convolving the error values with the error kernel. The adaptive pixel values for the pixels in column 2230 are calculated by multiplying the correction values by a sharpening factor,  $\alpha$ , and adding the product to the initial values.

After the adaptive pixel values of column 2230 are calculated, the simulated pixel values for the pixels in a column 2232 are calculated by convolving the initial pixel values with the appropriate simulation kernel. Error values are generated for each pixel in column 2232 by subtracting the simulated pixel values from the original image pixel values. The correction values for the pixels in column 2232 are calculated by convolving the error values with the error kernel.

The final sub-frame pixel value corresponding to pixel 2212 is calculated using the values generated by the above algorithm, the history values 2222, the error values 2224, and a sharpening factor,  $\alpha$ .

Intermediate calculations used in calculating the final sub-frame pixel value corresponding to a given pixel may be re-used in calculating the final sub-

frame pixel value corresponding to a pixel adjacent to the given pixel. For example, intermediate calculations used in calculating the final sub-frame pixel value for pixel 2002 may be re-used in calculating the final sub-frame pixel value of a pixel to the right of pixel 2002. As a result, certain redundant calculations 5 may be omitted.

The sharpening factor,  $\alpha$ , in the above algorithm may be different in calculating values of different columns using the center adaptive multi-pass algorithm with history. For example, the sharpening factor,  $\alpha$ , may be "3" for calculating the adaptive pixel values of column 2228, "1.8" for calculating the 10 adaptive pixel values of column 2230, and "0.5" for calculating the final sub-frame pixel value corresponding to pixel 2212.

Although the above algorithm was described for three iterations of the center adaptive multi-pass algorithm, the algorithm can be expanded or reduced to apply to any number of iterations by increasing or decreasing the number of 15 columns and the number of pixels in each column used in the above algorithm in accordance with the region of influence for the number of iterations.

Figure 32 is a block diagram illustrating the simplified region of influence 2210 of the pixel 2212 with respect to an image 2300 for three iterations of the center adaptive multi-pass algorithm with history where the final sub-frame pixel 20 values are calculated in a raster pattern. In Figure 32, the final sub-frame pixel value corresponding to pixel 2212 is calculated using the pixel values encompassed by the region of influence 2210 as just described in the above algorithm. To calculate the final sub-frame pixel value corresponding to a pixel 2302, the region of influence 2210 is shifted by one pixel to the right (not shown) 25 as indicated by the arrow 2304. Similarly, the region of influence 2210 is shifted down by one pixel (not shown) as indicated by the arrow 2308 to calculate the final sub-frame pixel value corresponding to a pixel 2306.

In other embodiments, the center adaptive multi-pass algorithm with history may be used to generate four sub-frames for four-position processing. 30 The four sub-frames are displayed with display device 26 using four-position processing as described above with reference to Figures 3A-3E. With four-position processing, the simulation and error kernels each comprise a 3x3 array

with the first row of values being "1/16", "2/16", "1/16", the second row of values being "2/16", "4/16", "2/16", and the third row of values being "1/16", "2/16", "1/16". In addition, a row of error values separate from a row of history values is used in the algorithm described above.

5 In other embodiments, the error kernel of the center adaptive multi-pass algorithm with history may be omitted. In these embodiments, the row of error values is not stored as shown in Figure 31, and the simulation kernel comprises a 3x3 array with the first row of values being "1/16", "2/16", "1/16", the second row of values being "2/16", "4/16", "2/16", and the third row of values being "1/16", "2/16", "1/16". With these modifications, the center adaptive multi-pass algorithm with history may be implemented in a manner similar to that described above for the simplified center adaptive multi-pass algorithm with history.

10

15 In one embodiment of sub-frame generation unit 36 (shown in Figure 29), sub-frame generation module 2110 implements the center adaptive multi-pass algorithm with history. In another embodiment, sub-frame generation unit 36 comprises an ASIC that implements the center adaptive multi-pass algorithm with history.

20 Embodiments described herein may provide advantages over prior solutions. For example, the display of various types of graphical images including natural images and high contrast images such as business graphics may be enhanced.

25 Although specific embodiments have been illustrated and described herein for purposes of description of the preferred embodiment, it will be appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent implementations may be substituted for the specific embodiments shown and described without departing from the scope of the present invention. Those with skill in the mechanical, electro-mechanical, electrical, and computer arts will readily appreciate that the present invention may be implemented in a very wide variety of embodiments. This application is 30 intended to cover any adaptations or variations of the preferred embodiments discussed herein. Therefore, it is manifestly intended that this invention be limited only by the claims and the equivalents thereof.